

# 开发月刊

Development Monthly

2012年07月

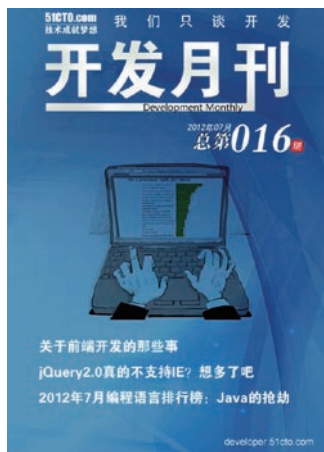
总第016期



关于前端开发的那些事

jQuery2.0真的不支持IE? 想多了吧

2012年7月编程语言排行榜: Java的抢劫



	<b>编程排行</b> <small>Billboard</small>
3	7月编程语言排行榜: Java的抢劫
	<b>专题报道</b> <small>《HTML 5开发》</small>
5	项目团队需要从新认识架构师的职责
7	一个很糟糕的项目是怎么炼成的?
8	Team Leader你会带团队?懂合作吗?
	<b>技术热点</b> <small>Techlogy hot</small>
10	Facebook如何编程:扎克伯格亲把关
12	百度王劲:用云计算更好服务开发者
13	jQuery2.0真的不支持IE? 想多了吧
15	前端开发里并没有多少优秀老开发人员
16	Yahoo前端优化十四条军规
19	我在谷歌的一次有趣的面试经历
21	King.com产品经理谈HTML5发展
23	非小型电子商务系统设计经验分享
25	腾讯发布《Web App白皮书》
26	程序员怎么提高英语阅读水平
27	两天完成一个小型工程报价系统
28	交互设计那些事儿:开发人员必备知识
31	Python语言十分钟快速入门
34	软件项目管理流程总结
37	多线程程序与单进程程序的架构
38	如何写一份良好的缺陷 ( Bug ) 报告
40	我对架构师的理解:如何成为一个合格的架构师
41	关于前端开发的那些事



# 7月编程语言排行榜：Java的抢劫

TIOBE 今天公布了 7 月份编程语言排行榜, C 语言和 Java 之间的差距正在加大, 已经超过 3%。而 Objective-C 已经悄然升到老三的位置, 压倒了之前的 C++。而 PHP 正逐渐下滑, 只能守住第 7 的位置。

TIOBE 今天公布了 7 月份编程语言排行榜, C 语言和 Java 之间的差距正在加大, 已经超过 3%。而 Objective-C 已经悄然升到老三的位置, 压倒了之前的 C++。而 PHP 正逐渐下滑, 只能守住第 7 的位置。

以下是前 20 名编程语言排行:

Position Jul 2012	Position Jul 2011	Delta in Position	Programming Language	Ratings Jul 2012	Delta Jul 2011	Status
1	2	↑	C	11.011%	+1.05%	A
2	1	↓	Java	10.087%	-3.16%	A
3	6	↑↑↑	Objective-C	9.335%	+1.15%	A
4	3	↓	C++	9.118%	-0.10%	A
5	4	↓	C#	6.768%	-0.45%	A
6	7	↑	(Visual) Basic	5.595%	+0.59%	A
7	5	↓↓	PHP	5.012%	-1.17%	A
8	8	=	Python	4.000%	-0.42%	A
9	9	=	Perl	2.053%	-0.28%	A
10	12	↑↑	Ruby	1.768%	0.44%	A
11	10	↓	JavaScript	1.454%	-0.79%	A
12	14	↑↑	Delphi/Object Pascal	1.157%	-0.27%	A
13	13	=	Lisp	0.597%	-0.09%	A
14	15	↑	Transact-SQL	0.554%	-0.15%	A
15	25	↑↑↑↑↑	Visual Basic .NET	0.517%	-0.43%	A
16	16	=	Prolog	0.337%	-0.17%	A
17	19	↑↑	Ada	0.589%	-0.14%	B
18	11	↓↓↓↓↓	Fortran	0.464%	-0.89%	B
19	24	↑↑	PL/SQL	0.145%	-0.10%	A-
20	26	↑↑↑↑	MATLAB	0.539%	-0.18%	B

## Java 的劫数来了?

看到这样的榜单, 我们该感叹为何 Java 会有如此的下场。尽管与身后的 Objective-C 还有 7 个百分点的差距, 但已经尽显疲态。按照 Android 系统的市场占有率, Java 的表现不应该如此。Java 果真不行了吗? 我们先看一段视频。

视频地址:

<http://developer.51cto.com/art/201207/346386.htm>

JavaZone 似乎是要把它作为一个传统, 每年都制作一个关于 Java 的电影宣传片, 比如 2010 年的《Java 风云》以及《Lady Java》, 2011 年的恐怖大片《Java 惊魂》, 当然, 现在又是出大片的时候了, 今年的片子叫做《Java 打劫 (The Java Heist)》, 不知道是模仿的那部片子, 还是挺有趣的。

在这部片子中, 一帮劫匪计划去打劫威斯敏斯特银行。自以为在银行金库主机完成转账就能高枕无忧。大家戴着 Anonymous 标志性的笑脸面具冲进银行, 却发现搞不定 Java 而被捕。最后劫匪头子在监狱里潜心研究 Java, 样子有点像 Java 之父高斯林。

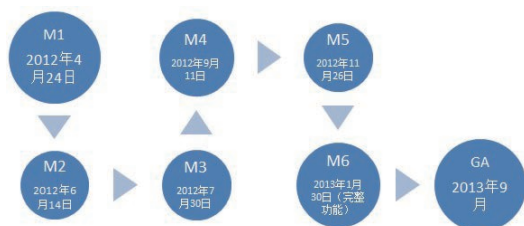
从这个 Java 粉丝拍的片中, Java 似乎是不可攻陷的堡垒。尽管他是开源的, 免费的, 但这一点都不妨碍它的高可用性和高安全性。其实对于开发者, 如何精通一门语言, 比任何排行榜都重要。

虽然很多程序员对于 Java 语言被 C 语言超过的消息不屑一顾。但我们从 Oracle 收购 SUN 之后的表现看, 似乎 Oracle 只是把 Java 当作一颗摇钱树。透过 Java, Oracle 可以伸手找 Google 要专利费, 甚至对簿公堂。但对于 Java 的未来, Oracle 真的很关心吗?

## 下一代标准: Java 8

在 2012 年 4 月, Oracle 发布了 Java 8 的路线

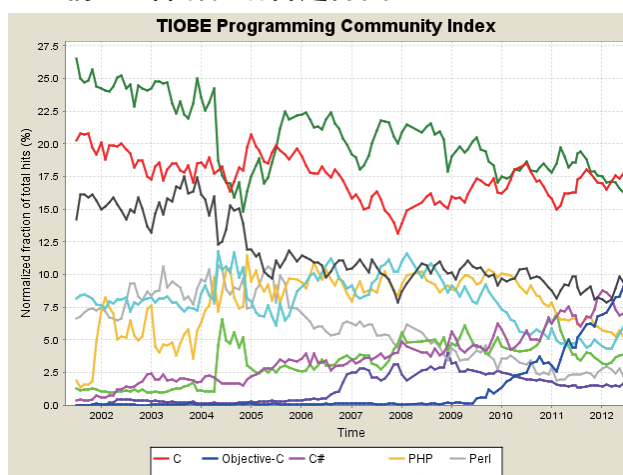
## 2012 年 7 月编程语言排行榜: Java 的抢劫 II



那么根据这个时间表,大家很快就能见到 Java 8 了。比如 Jigsaw, 该项目的目标是创建在 JDK 设计和实施模块系统有用的方式,然后应用该系统到 JDK 本身。这一点使处理大块代码更易于管理,并促进跨应用程序的代码复用和更好的维护。Jigsaw 项目将带来许多新的功能涉及封装,重构,版本和模块集成。

在 JavaOne 的大会上, Oracle 还讨论了 Java 9 候选发展的关键领域。他们正在寻找加入一个自我调整的 JVM,改善本地集成和大规模多核的可扩展性。他们还着眼于通过添加新的元对象协议实现跨语言支持和云应用资源管理。下面是本期编程语言排行榜的其他排名数据和趋势走向。

## 前 10 名编程语言走势图



## 下面是第 50 到 100 的编程语言排名

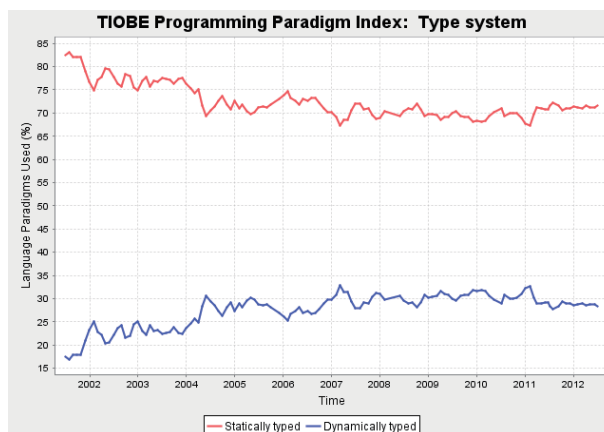
(Visual) FoxPro, ABC, Algol, Alpha, Arc, ATLAS, Avenue, Awk, Bash, bc,

Beta, Boo, Bourne shell, CFML, cg, CL (OS/400), Clean, cT,

Dylan, Eiffel, Factor, Groovy, Inform, Io, J, JavaFX Script, Korn shell, LabVIEW,

MAD, Magic, Maple, Mercury, MUMPS, NXT-G, Oberon, Object Pascal, Objective Caml, Occam, OpenCL, Oz, PILOT, PowerShell, Prolog, Revolution, S, Scala, Smalltalk, Spark, Standard ML, Tcl

## 下面给出了编程语言类别的一年变化趋势



Programming Language	Position July 2012	Position July 2007	Position July 1997	Position July 1987
C	1	2	1	1
Java	2	1	5	-
Objective-C	3	46	-	-
C++	4	3	2	6
C#	5	7	-	-
(Visual) Basic	6	4	4	5
PHP	7	5	-	-
Python	8	8	23	-
Perl	9	6	6	-
Ruby	10	10	-	-
Lisp	13	15	16	3
Ada	17	16	12	2
COBOL	30	17	3	11

上图为长期以来,各编程语言位置变化。我们可以看到 C 语言其实比 Java 更长时间占据编程语言排行榜。■

# 项目团队需要重新认识架构师的职责

成为一个优秀的架构师还有很长的路要走  
(软件架构案例分析和最佳实践培训收获)

2009-12-25 到 27 日我们参加了某软件培训机构的《软件架构案例分析和最佳实践》课程培训,开拓了眼界,收获很多,刘老师讲得不错,非常有实战经验,跟他学到了不少有关软件架构的知识,可惜的是 3 天的培训课程不可能完全掌握所有知识,师傅只是给我们打开了一扇门,指出了方向,成为一个优秀的架构师还有很长的路要走。

新视野 “软件架构” 定义的决策因素

## 定义 1: 架构是一系列重要决策的集合

一直以来,学习架构,使用架构,关注点都仅限于技术层面,没有认识到架构和“决策”的关系,这说明架构是一个很重要的概念,从软件架构概念产生的背景可以得出:

其实,软件架构 (Software Architecture, 软件体系结构) 一词早在 20 世纪 60 年代就被 E.W.Dijkstra 提出,但是直到 20 世纪 90 年代初才开始流行起来。为了提高软件需求和软件设计的质量,软件工程界提出了需求分析工程技术和各种软件建模技术。但是在需求和设计之间仍然存在一条很难逾越的鸿沟,即缺乏能够反映做决策的中间过程,从而很难有效地将需求转化为相应的设计。为此,软件架构概念应运而生,并试图在软件需求与软件设计之间架起一座桥梁,着重解决软件系统的结构和需求向实现平坦过渡问题。

定义 2: 软件架构为软件系统提供了一个

结构、行为和属性的高级抽象,由构成系统的元素的描述、这些元素的相互作用、指导元素集成的模式以及这些模式的约束组成。

软件架构不仅指定了系统的组织结构和拓扑结构,并且显示了系统需求和构成系统的元素之间的对应关系,提供了一些设计决策基本原理。

还有很多其它的定义方式,但从这两个定义可以看出,架构对于决策的重要性,架构师的工作对于项目的成功运作具有决定性的作用。

“架构师”不是空头衔,不是项目经理,开发人员,测试人员的兼职角色。

在软件工程领域中,软件架构师实际上就是软件项目的总体设计师,是软件组织新产品的开发与集成、新技术体系的构建者。Martin Fowler (著名软件架构和设计大师,软件设计模式创始人)指出:

架构师是对所有重要事情做出决定的人。

软件架构师在整个软件开发过程中都起着重要作用,并随着开发进程的推进而其职责或关注点不断地变化。

在需求阶段,软件架构师主要负责理解和管理非功能性系统需求,比如软件的可维护性、性能、复用性、可靠性、有效性和可测试性等。此外,架构师还要经常审查客户和市场人员所提出的需求,确认开发团队所提出的设计;在需求越来越明确后,架构师的关注点开始转移到组织开发团队成员和开发过程的定义上。

在软件设计阶段,架构师负责对整个软件架

## 项目团队需要重新认识架构师的职责 II

构、关键构件、接口的设计。

在编码阶段,架构师则成为程序员的顾问,并且经常性地要举行一些技术研讨会、技术培训班等。

随着软件开始测试、集成和交付,集成和测试支持将成为软件架构师的工作重点。

在软件维护开始时,软件架构师就要开始为下一版本的产品是否应该增加新的功能模块进行决策。

软件架构视图

——软件架构是一种无法以简单的一维方式进行说明的复杂实体。

——多重软件架构之所以必不可少,是因为各类涉众(用户,客户,开发人员,测试人员,维护人员,内部操作人员,其他人员)需要从各自的角度理解和使用架构。

常用的软件架构视图:

功能视图

开发视图

进程视图

部署视图

场景视图

数据视图

实现视图

注:在我们的实际项目中,用的最多的是功能视图,其次是开发视图,没想到还有这么多的视图需要考虑。比如,在 MB 一期的设计中,我曾考虑过是否有必要作一个软件的部署形式图,最后犹豫中还是出了一个,现在看来是很有必要的了,至少让运维人员明白了 MB 的软件部署是怎么回事。

新观念

### 架构的质量属性

在现实的系统中,决定系统成功或者失败的关键因素中,满足非功能需求往往比满足功能性需求更为重要。从技术角度看,质量属性影响重要的架构和设计策略。

质量属性分为系统质量属性和商业质量属性,其中系统质量属性又分为运行时期的质量属性和开发时期的质量属性;商业质量属性包括政治因素,上市时间,成本和收益等。

我们虽然常常把性能,安全,可扩展等词挂在嘴边,但往往在实际开发中这些因素都忽略了,为了赶工期,功能实现是第一位的,最后软件做出来了,质量却不好,问题一堆。实际上,软件的质量不只是产品经理应该关注的,软件架构师也必须关注,给出建议,供管理层做出决策。MB 的开发就是最明显的例子,上头规定了上线时间,满足必须的功能,及时上线是附在开发人员身上的魔咒,开发人员只得加班加点的工作,最后软件及时上线了,但在运行效率,易用性等方面成为诟病。

### 架构是有生命力的

运维人员说:软件运行这么慢,架构太烂了!

开发人员:代码这么难写,架构太不灵活了!

客户:软件太不稳定了,架构有没有问题啊?

XXX 说:YYY 架构师太差劲了,怎么就没有设计出一个好架构?

在所有人看来,架构必须是完美的,对所有人感觉都是良好的,能够适应未来的种种变化,能够一劳永逸!起初我也是这么认为,但老师告诉了我们一个新观点。■(本文未完,更多访问链接)

<http://developer.51cto.com/art/201206/344330.htm>



# 一个很糟糕的项目是怎么炼成的？

一个很糟糕的项目是怎么炼成的？那么自己在项目里看到的，想到的，自己做了下总结，希望对以后做项目的人有帮助。



首先本人是个小程序员，在 10 个人左右的团队里面，为了完成今年 5 月份第一期的交付。整个项目的负责人兼职需求吧，需要做手术然后就离开的大家，另一个需求人也离开了团队，这样的话就直接交接了 2 天，就匆匆离开了团队，虽然我们不知道这是个很大的风险点，但没办法这是领导的决定。以下是自己在项目遇到的没解决的问题：

## 1. 糟糕平台，在项目里面放了 N 个临时炸弹

同样，由于上面决策的原因，整个项目依赖一个在与我们项目并行开发的平台，你可以想象下，平台发布版本的频率直接影响项目的进度，甚至成了项目的绊脚石，可以这么说，你今天跑起来的代码，明天早晨来了之后，可能就跑不起来，这个问题，足以让你去摔鼠标了。

2. 做平台只有一个人在上海做，大量的平台 BUG 出现了。开发平台的人员在上海，并且没有测试人员，平台必须在项目里面进行试验，但是他在上海，所有的问题都不知道是谁的错，平台踢到项目，项目踢到平台，因为大家都不知道这些问题该有谁来做。

## 3. 计划会议被干掉了。

这样折腾了 1 个迭代，马上就要到第一期交付了，做的刚到整个项目的百分之三十，所以，就把计划会议干掉了，这可是个糟糕的决定。

## 4. 测试人员紧缺

9 个开发，一个测试，测试人员不到位，由于

计划会议被干掉，大家交叉测试的能力都没了。

## 5. 每天立会都吹锋号，加班成家常便饭了

天天加班，造成整个团队的战斗力下降到了极点，大家都在拼命的赶自己任务，由于用到的 WCF, SILVERLIGHT, 以及 linq, 大家做事都畏手畏脚的，加上平台放的临时炸弹，抱怨平台，抱怨需求，抱怨时间，抱怨目标太高，可以我们大家都知道，必须的完成交付。

## 6. 没有美工

这个真的是很糟糕的一个决定，一个列表的线有上面左边，没右边，还有很多图标。

## 7. 我是按原型做的“

需求是没人研究的，现在几乎是按原型设计，但是里面真的很多地方不通，由于时间的原因大家都安原型去做，但是原型本身设计都很糟糕，这是最致命的一点。

## 8. 平台的特点：最后还要合代码

平台的特点每个人开发自己自己小模块，最后还要合各个模块，这个合并也很耗费时间。第一期交付，项目是见用户了，项目负责人回来说很好，我当时很诧异。

总结：项目时间真的好紧张，平台真的很糟糕，业务有很多盲点，自己新技术也是刚刚接触，这些盲点有时候真的会把我们搞死的。希望以后这些盲点可以控制好。虽然很多问题，我不知道怎么去解决，但是我想以后我肯定会解决的。■

# Team Leader你会带团队?懂合作吗?

团队管理是一个比较大的范围和概念,但我们可以把它进行简化到以团队为基础,在团队上进行一些方法的应用。我在文章中,将分为不同的块讲解。当你把这些不同的块都理解清楚,结合起来就是团队管理。

PS: 文章中的一些理解,是基于我学习一些管理书籍的内容和在工作中实践总结的一些个人概念的叙述。是一种经验的分享,可能会包含错误和不全面。需要读者自己去判断和理解。

为了能让大家对团队更好的理解,我讲的很细,导致了字数严重偏多。我本来想一起放上来,但怕大家看着睡觉,故分成了上下两部分。下篇会重点集中在团队的内部培训和团队成员的招聘,还有团队关系处理上。先把上篇放出来看看大家的意见和反响,也可方便我对下篇进行调整。看完后,如果你觉得受用,请推荐下。共同提高,我们的工作和发展才能更和谐。以团队开头:(如果你知道什么是团队,可以跳过)

什么是团队? 这里有个比较标准的解释: 团队( Team )是由员工和管理层组成的一个共同体,它合理利用每一个成员的知识和技能协同工作,解决问题,达到共同的目标。

在软件公司,一个基本的团队单位,是由 Team Leader (也可能是项目经理,每个公司对头衔的分配不一样。这里以 Team Leader 为总的称谓)所带领的一群人(如程序员)。以数据结构来说,就是一个多叉树的结构;每一个节点都是下面子节点的领导;节点和子节点组成了一个团

队。Team Leader 的领导(项目主管或项目经理)又会管理着多个 Team 子节点(开发组,测试组,美工组等);这些 Team 组成了一个大的 Team 团队,主管也是大 Team 的 Leader。换句话说,每个节点都是下面子节点的 Team Leader,至于起什么头衔,并不会影响 Leader 的职能。

团队类型和选择:(重要,直接决定着后续多个因素)

团队的类型其实没有一个什么固定的称谓和标准。一些团队领导在工作中不断调整团队成员的组成。当达到一个比较成熟的团队模式时,为了方便大家理解,会拿生活中大家熟知的一些事物来代替(如手术台类型)。

其实,前人已经为我们总结出了很多成熟的团队组成类型。如:手术台类型。做手术大家一般比较熟悉,是以一个主治医师为主刀,其他医生为辅助,来完成整个手术。我们可以很容易的把这种模式与团队组成结合起来。一个团队中,以一个或两三个程序为主导,其余程序员为辅助来完成代码开发;这样的模式在国内游戏开发中经常使用。主刀也就是我们常说的主程(主要程序员或技术一把手),以主程为核心进行开发,其他程序员辅助完成附加功能。

在实际的工作中,我个人倾向于交响乐队类型(这个是我个人的叫法,因为这个模式和交响乐队基本一样,也便于大家理解。我在后面的“团队合作”里进行了较详细说明和对比)。在交响乐团中,团员都有各自的技能,并分工明确;在指



## Team Leader 你会带团队吗？你懂合作吗？ II

挥的指导下完成演奏。这里的指挥就是我们的 Team Leader。

团队的类型有很多,这里只是列了两个被大家广泛使用的类型;其它的类型,就请大家自己钻研了。

选择什么样的团队类型,是一个 Team Leader 在组建团队时要解决的一个问题。不同的团队类型涉及不同的团队成员组成,并对团队成员技能的要求也不一样。这会涉及到你需要招什么样的人,这也是为什么我把副标题定为:打好你手中的牌。不同的团队成员就如同不同的扑克牌,你要去组织和打好它。发散下思维,可以把团队管理抽象成打扑克牌的过程,打牌就是在进行管理。

不同的团队类型有不同的优劣。如手术台类型,对主程的要求很高。在整个团队开发中,主程的个人能力决定了整个项目的成败;在很多时候,主程也担当着 Team Leader 的角色。主程个人能力这个因素,导致了我不太喜欢使用这个模式;因为风险较集中。但是在一些尖端技术的开发中,往往又是需要手术台这种模式。原因很简单,尖端当然需要顶尖人才去做了。这个模式有一个好处,就是对其他程序员的要求不高,比较容易招到合适的人。

再来说说交响乐队类型。这个是我普遍推荐的,很大一部分原因是我们并没有很尖端的技术要去实现,并且这个模式可以分散风险。但是这个模式对程序员的能力有一定要求,就是需要某方面的特定能力(具体需要什么能力,根据你的项目需要来划分)。我不需要你很多东西,你只需要对特定领域有钻研;或者你对某个领域感兴趣,并愿意往这个领域走下去。我需要你专精一

点,如同交响乐队里面每个人有各自的乐器技能。但这模式也有个缺点,那就是在初级程序员的招聘上,基本很难找到有钻研一个方向的。毕竟他们对程序才起步。不过这个问题通过内部培训是可以解决的。后面会具体探讨内部培训。

所以,选择什么团队类型要看项目需要和 Team Leader 的经验。发散下思维,其实在好多小公司或非正规团队,也是使用的手术台类型。3-5 人一个团队就上架干活了,1 个技术领头的(主程),搭配 2-3 个人就干了。这种类型好搭建,成本低。换作交响乐队类型,同等情况一般人员配置会稍多,成本稍高。但是作为长期发展来看,交响乐队类型的团队比较稳健和可持续发展;如果是要技术难点突破,手术台类型的团队就比较适合去攻克难题。

提点敏捷开发,敏捷开发也是一种团队类型,被大家尝试过、讨论过、分析过。我也尝试过,最终还是转回了交响乐队的模式。这里简单说下我对敏捷开发的拙见。优势,快速开发,快! 非常快! 劣势,按照敏捷开发的理论,应该是没啥劣势。一切看起来是那么美好的敏捷开发,到具体操作起来却很多失败。其主要问题就是,敏捷开发对开发团队的人员平均素质要求太高,一般的中小公司,很难达到这个平均素质水平。这也是我失败的原因。

团队合作:(别跟我说你会团队合作,十有八九你在忽悠)

团队合作根据团队类型的不同,合作的方式也是不同的。■

本文未完,完整部分请浏览

<http://developer.51cto.com/art/201206/343599.htm>

## ■ 编者按

究竟 Facebook 是如何编程的,身为 CEO 的马克·扎克伯格对 News Feed 每个代码更新亲自把关,确保不出现任何差错。

# Facebook如何编程:扎克伯格亲把关

美国科技博客网站 BusinessInsider 今日撰文揭开了 Facebook 程序设计人员的神秘面纱。Facebook 代码从开始编写到最终发布,都有着极为严格的程序,CEO 马克·扎克伯格 (Mark Zuckerberg) 甚至对 News Feed 每个代码更新亲自把关,确保不出现任何差错。以下为文章概要:

Facebook 工程师是这家社交网站巨头最有价值的财产,他们拥有非常大的自主权,但同时也面临着尽快发布高质量代码的压力。

谷歌员工李易 (Yee Lee) 通过与大批 Facebook 工程师的交谈,在一篇博文中披露了 Facebook 代码发布流程。这个流程的确与外界之前想象的相差无几,虽然 Facebook 对代码发布的监督比人们预想的更为严格。尽管这篇博文发表于一年前,但整个流程自 Facebook 上市以来似乎并未发生太大变化

Facebook 员工主要由工程师构成,人数最多的两个团队分别是 Engineering 和 Ops,总计占了 Facebook 员工总数的一半左右。此外,Facebook 还有大量产品经理。他们要确保代码按时发布。每一个产品经理负责 7 到 10 名工程师。

Facebook 所有工程师都要接受 4 到 6 周的培训,学习 Facebook 修复漏洞的方法,聆听资深雇员举办的讲座。在进入“训练营”(Boot Camp)的工程师当中,会有大约 10% 无法顺利通过测试,最终被劝退。

接受完培训以后,工程师开始接触 Facebook 数据库。他们可以随意核对代码,对数据库做出修改。员工们还会拿到一张“禁做之事”名单,如禁止分享用户数据。尽管如此,Facebook 还出台了一系列措施,防止此类事情的发生。

同谷歌一样,Facebook 的企业文化同样以工程师为主。一名工程师说:“产品经理基本上在这里毫无作为。”工程师可以修改尚未正式上市的产品规格,在任何时间提出新的功能创意。

工程师会在每月一次的不同团队例会上提交他们的成果。产品推广经理和产品经理会出席这些会议,但不被鼓励畅所欲言。“如果产品经理在例会上畅所欲言,工程师们就会向领导层反应说,‘上次会议上他们有关产品的意见太多了’。”

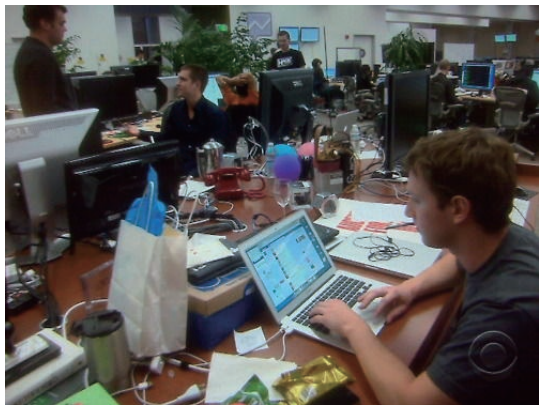
在 Facebook,工程师想做什么,基本上都由他们自己决定。他们会找到主管,说:“这是我想做的五件事情。”产品经理会说服工程师当场试一试,让他们亲身体验这些项目的效果,但他们多数情况下不会对每位工程师的偏好横加干涉。

工程师们不会争论某项功能是否值得尝试,而是开发出原型机。接下来,工程师会用一周的时间开发某项功能并进行测试,以确定它是否值得推出成品。通常情况下,新功能都是由 Facebook 员工亲自测试。整个过程由一款名为“Gatekeeper”应用控制。这是 Facebook “黑客”文化的主要组成部分——快速开发和推出产品,

## Facebook 如何编程：扎克伯格亲把关 II

淘汰没有市场前景的产品。

在 Facebook,每个人都想参与后端产品的开发。可伸缩性和基础架构是工程师最感兴趣的两个问题。所以,工程师很难对实时消息等前端产品感到兴奋,相反,每个人都希望从事像 News Feed 算法这样的后端产品。这种做法与其他消费类科技企业的惯例背道而驰,在这些企业,员工都希望参与前端产品的研发工作。



Facebook 创始人兼 CEO 马克扎克伯格 (Mark Zuckerberg) 会亲自对 News Feed 每个代码更新把关。在 Facebook,所有重大升级的代码都进行强制评估,任何一个改动都至少由一人把关。但是,无论工程师对 News Feed 做出任何改动,都将由扎克伯格亲自把关。

Facebook 工程师负责测试产品功能,修复产品漏洞,对发布以后的产品进行维护,但他们并不是官方的质保团队。不过,Facebook 仍然有负责质量评估的工程师,并积极鼓励每位工程师报告产品漏洞。

正常情况下,代码升级会在每周二发布。Facebook 有专门的评估工具,告诉工程师代码更新的风险有多大。

运营团队会逐步推出代码更新。Facebook

共有大约 6 万台服务器,运营团队会逐步将更新后的代码发布到少数几台服务器上,确保它能起作用。最开始是 6 台服务器,接着慢慢增加。如果需要做出修改,那么这项工作会由提交代码更新的工程师在线下完成。修复工作完成后,代码会再次在那 6 台服务器上先试用,接着增加到更多的服务器上。

在代码更新发布期间,运营团队会通过 IRC 和其他实时聊天工具一对一通知工程师,他提交的代码是否需要修改。如果修复以后的结果仍然难以令运营团队满意,当事工程师会被“当众羞辱”,虽然李没有提供有关这方面具体细节,但他说如果这种情况经常发生,工程师会被炒鱿鱼■



下载通道: <http://os.51cto.com/art/201207/348118.htm>

《Linux 运维趋势》是由 51CTO 系统频道策划、针对 Linux/Unix 系统运维人员制作的一份开放的电子杂志,内容从基础的技巧心得、实际操作案例到中、高端的运维技术趋势与理念等均有覆盖。我们的所有内容均收集整理自国内外互联网,每篇文章都会严格标注出处与作者,同时编辑也会尽力联系每一篇文章的原作者进行确认。



# 百度王劲:用云计算更好服务开发者

2012年6月21日下午,上海正在经历大雨之前的宁静。这座作为中国金融中心的城市,住在这里的开发者会有什么样的技术需求?

百度开发者大会上海站的签到时间是下午1点半,不过来自上海爱立信的洛先生早早的就来到了会场。身为系统工程师的他,这次来主要是关注云计算方面的内容。除了自己对云计算感兴趣之外,更多的是希望能够跟百度的工程师做面对面的交流。从中得以知道百度在云计算方面有过什么样的尝试,在众多云计算方案中究竟是怎样做的技术选型。云计算成为了广大参会者最感兴趣的话题之一。

## 云计算和大数据成焦点

而云计算也是百度技术副总裁王劲先生在本次活动中演讲的主题。在演讲的一开始,王劲先生就提到了互联网企业正面临大数据的难题,如此庞大的数据该如何快速处理和分析?另一个就是随着移动互联网的快速发展,出现了多终端、多应用的现象,数据散落在这些应用和终端中,开发者面临着数据碎片化的困境。我们该怎么办?

对此,王劲先生谈到解决这些问题的方法:可以把它放在云上,把这些数据统一在一起,就能够很好地对它进行数据挖掘和分析,很多的潜能也会被释放出来。开发者也可以更好分析和利用由不同设备造成的数据碎片,避免碎片化问题。

在随后的媒体专访环节中,51CTO记者还向王劲先生提到了大数据方面的问题。正如Facebook的Timeline、Google的BigQuery等等应用一样,互联网企业在大数据方面充当了急先

锋的角色。那么百度在大数据这一块,是不是会有相应的动作和计划?

王劲副总裁表示,搜索引擎及搜索广告引擎就是大数据的典型应用。比如百度的广告引擎拥有一个机器学习系统。通过将关键字拆分成百亿个特征向量,进行分析、匹配,最终得到满意的搜索结果。这一个数据分析与数据挖掘的过程,没有一定量级的数据是根本无法完成的。这里就用到了大数据的力量。

那么今后百度会通过自己在大数据方面的经验和技能积累,透过云计算这样先进的技术手段,向开发者推出更多的工具和分析方法,帮助大家更好的利用大数据的资源。

## 百度用云计算更好地服务开发者

云计算对于开发者来说能起到降低开发成本的作用。但对于服务提供商来说,数据中心的建设和维护,必然是一项庞大的开支。王劲就提到百度数据中心的服务器一台就耗资数万,那成百上千台服务器的消耗就不是一个小数字。为什么百度会愿意投入如此巨大的成本呢?归根结底还是为了更好地服务开发者,特别是透过云计算更好地服务开发者,开发出更好的应用。

对广大开发者而言,百度云是一种实力的分享。搜索引擎本身就是云计算最佳应用之一。■

本文未完,更多部分请访问

<http://developer.51cto.com/art/201206/344951.htm>

## ■ 编者按

总体来说。通过采访和统计网上的留言来看,真是喜也工程师,忧也工程师。开发者对于IE既痛恨又无奈,没办法,谁让我们是工程师,我们做的就只有找办法解决这一问题,既然不能改变,那么就去适应吧。

## jQuery2.0真的不支持IE? 想多了吧

网上放出很多关于 jQuery 2.0 将不再支持 IE 6/7/8 的消息,这个消息一放出真是有人欢喜有人忧。喜得是开发人员,终于可以脱离 IE 这让人头疼的浏览器,头疼的其实也是程序员,很多程序员所开发的东西就是为了服务 IE 而产生的。

关于这条消息,网上的评论很多,有人说让 IE 早点灭亡, jQuery2.0 将追随 HTML 5 的脚步。也有人说这是推动 Web 向前发展的好消息, IE 现在可能是 Web 的最大阻力。还有些人说 jQuery 的存在价值就是跨浏览器兼容,减轻前端工作,不兼容浏览器只是玩火自焚。

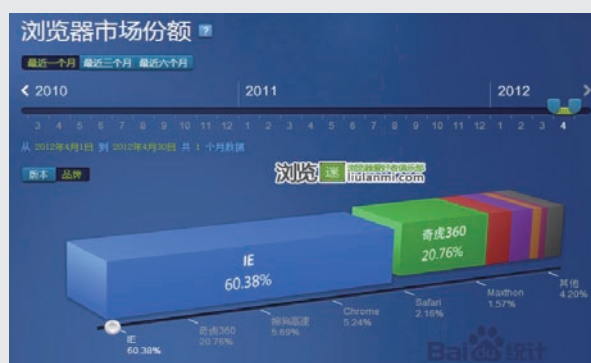
不过这条消息很快被得到否定, jQuery2.0 是说不再支持 IE6/7/8 中特有的功能,比如 borked event model, IE7 “attroperties”, HTML5 shims 等。所不支持的只是一些特性而已,并不等于不支持浏览器,并且还可以加一段代码解决这一问题。

### 如果 jQuery2.0 真的不支持 IE6/7/8

我们想一下,如果 jQuery2.0 真的不支持 IE6/7/8,这肯定是一个非常疯狂的决定。那么将会是 jQ 抛弃了中国呢还是中国抛弃了 jQ 呢? 国内使用 IE 的用户还是大多数,并且很多公司的老板用的依旧是 IE。当他们开启 IE 页面发现不兼容时,他们第一个想法会是什么? 一定不会是换个浏览器去看,而是告诉手底下的开发人员:

“你给我把它改兼容了”。也许国内 IE 用户的数量多到你想象不到,下面我们来看一组数据。

下图为 2012 年 4 月份国内主流浏览器市场份额排行榜



从图中不难看出 IE 的比例是多么的可怕,与 2011 年的这个时候相比可以说并没有减少。再看一张图。

下图为国内浏览器市场浏览器版本占有率情况



上图中, IE6/7/8 依旧是主导地位,而深受开发人员喜爱的火狐却连一个标注都没有。

## jQuery2.0 真的不支持 IE? 想多了吧 II

以小编看来,想要 IE 下台的这种想法,无疑是痴人说梦。IE 是出现最早的浏览器,就算你如何说别的浏览器好,难道别人就会听从你的意见换掉已经用习惯的浏览器?

### jQuery2.0 不支持 IE6/7/8 开发者心声

据小编采访的 jQuery 开发者得到的信息。M 工程师所在的公司大概 200 人左右,他主要还是用 jQuery 的一些开源库插件这一块。当我询问他若是 jQuery2.0 真的不会在支持 IE 时,这位开发者冷静的回答了一句作为 jQuery 开发者,你总不能为了用 2.0 而放弃 IE 用户吧,并且他感觉 jQuery2.0 对 IE 造成的影响并不会很大,用户才是上帝,你不可能让用户去改变原有的浏览器,你能做的只是去适应用户。

在聊到如果真的可以解套 IE 时,他说那将会是他梦寐以求的事情。只要做 Web 开发的都会痛恨 IE。

造成这些的原因有:浏览器适配的问题、IE 还有各种解决起来让人非常头疼的 bug。如果项目涉及到浏览器比较少的时候还好说,涉及比较多的时候那么就是非让人头疼了。并且表示, jQuery2.0 出来以后也不会去立刻就用,还会停留在自己常用的版本。

小编随即又问了另外一个 jQuery 开发者,这个开发者是元器件交易网的陈工程师。陈工程师对于 IE 可见是相当的痛恨,他说他觉得这可以推动 Web 的发展,并且 jQuery2.0 出了你可以不用是不是。

若是 jQuery 有不向 IE 低头的这种气概就好了。不过他又说道,他觉得 jQuery2.0 不支持

IE6/7 倒是可以理解,IE8 都不支持就有点说不过去了。话说回来,这还是个梦啊。

要知道,公司服务的是客户,并不是程序员。正所谓市场决定一切,有一个视频我还记忆犹新,里面内容讲的是泰坦尼克号导演知道中国删减泰坦尼克号一部分内容时表现出的惊讶和不解,不过最后导演笑着释然了,并不是想明白了为什么删减,而是因为删减后的泰坦尼克号依旧创造出了很好的票房。

若 jQuery2.0 真的不支持 IE,带来的变动将会是浏览器的变革还是放弃 jQuery,我认为后者更大一些,也许更会有一个替代 jQuery 的产物出现。



### 总结

总体来说。通过采访和统计网上的留言来看,真是喜也工程师,忧也工程师。

开发者对于 IE 既痛恨又无奈,没办法,谁让我们是工程师,我们做的就只有找办法解决这一问题,既然不能改变,那么就去适应吧。

不过, jQuery 无疑和 HTML 5 一样,会给我们带来更美观的体验度,兼容是再好不过的。距 jQuery2.0 发布还有一阵子的时间,到时它会为我带来什么惊喜呢,让我们拭目以待吧。■



# 前端开发里并没有多少优秀老开发人员

## 编者按

如果你知道任何一个超过 40 的优秀开发人员,请做一件对你有益的事:向他请教。看看他掌握的那些技术是否还跟得上时代……

你知道有几个开发人员是超过 40 的? 对不少人而言,这个数是 0。如果你们当中知道一两个超过 40 开发者的人,我有个问题:他们中有几个人能称得上优秀? 如果你知道任何一个超过 40 的优秀开发人员,请做一件对你有益的事:向他请教。看看他掌握的那些技术是否还跟得上时代。一旦你做了这些,请大家做一件事,就是找个地方把他公布出来,因为实际上,在这个前端开发行业里并没有多少优秀的老的开发人员。

### 出现这种情况有两个原因。

第一:绝大多数的开发人员在工作了几年后都转向了管理岗位,一方面是因为他们原本的职业生涯规划就是这样(不管你喜欢与否,很多开发人员仍然认为,作为一个开发人员就该朝着这个方向奋斗),要么是他们因为偶然的原因进入了管理层。

对于后一种情况,我们只讨论那些优秀的开发者,他们表现良好,不管交给任何任务都能出色的完成。等进入管理层后,他们绝大多数的时间都是被锁在会议室里,根本没有任何时间剩余来做编程——尽管他们中的有些人很可能更喜欢把精力更多的放在编程上。

两种情况中,这些人很快都会失去开发者的身份,早晚的事儿。

第二个主要的原因是,很多具有较长阅历的开发人员慢慢的会认为自

己该掌握的知识都差不多掌握了,不再去学习新的解决问题的方法,或者是看看别的开发人员在学什么,自己也跟着学。

他们认为自己做过的那些就是一切,任何具有挑战的开发任务都处理过。他们一年又一年的使用着相同的技术和方法,心想着“为什么要去改变一个冠军组合?”。

虽然他们的技术方案可以用,也许会非常的好用,但他们的实现方法可能是过时的或效率低的。

从这点上看,程序员的品质开始降低,持续的降低,一直到他掌握的技术过期、被淘汰。



我不知道你会怎样,但我对在余下的职业生涯中走这两种道路是丝毫没有任何的兴趣。

我希望一直能这样编程编下去,我在其中找到了乐趣。(本文未完,更多请阅读原文)■

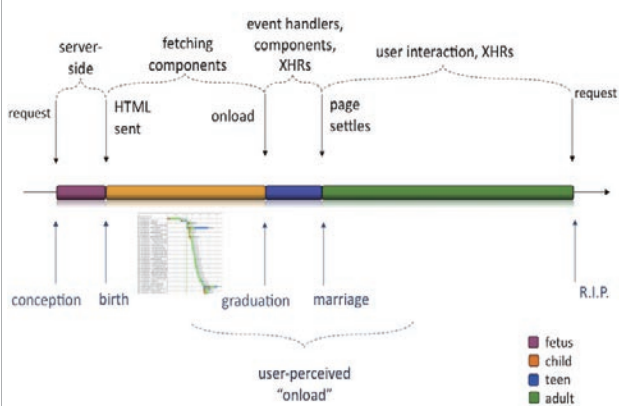
<http://developer.51cto.com/art/201207/347943.htm>

# Yahoo前端优化十四条军规

相信互联网已经越来越成为人们生活中不可或缺的一部分。Ajax, flex 等等富客户端的应用使得人们越加“幸福”地体验着许多原先只能在 C/S 实现的功能。比如 Google 机会已经把最基本的 office 应用都搬到了互联网上。当然便利的同时毫无疑问的也使页面的速度越来越慢。

相信互联网已经越来越成为人们生活中不可或缺的一部分。Ajax, flex 等等富客户端的应用使得人们越加“幸福”地体验着许多原先只能在 C/S 实现的功能。

比如 Google 机会已经把最基本的 office 应用都搬到了互联网上。当然便利的同时毫无疑问的也使页面的速度越来越慢。自己是做前端开发的,在性能方面,根据 Yahoo 的调查,后台只占 5%,而前端高达 95% 之多,其中有 88% 的东西是可以优化的。



以上是一张 web2.0 页面的生命周期图。工程师很形象地讲它分成了“怀孕,出生,毕业,结婚”四个阶段。如果在我们点击网页链接的时候能够意识到这个过程而不是简单的请求-响应的話,我们便可以挖掘出很多细节上可以提升性

能的东西。今天听了淘宝小马哥的一个对 yahoo 开发团队对 web 性能研究的一个讲座,感觉收获很大,想在 blog 上做个分享。

相信很多人都听过优化网站性能的 14 条规则。更多的信息可见 developer.yahoo.com

## 第一条、尽可能的减少 HTTP 的请求数 ( Make Fewer HTTP Requests )

http 请求是要开销的,想办法减少请求数自然可以提高网页速度。常用的方法,合并 css,js( 将一个页面中的 css 和 js 文件分别合并 ) 以及 Image maps 和 css sprites 等。当然或许将 css, js 文件拆分多个是因为 css 结构,共用等方面的考虑。

阿里巴巴中文站当时的做法是开发时依然分开开发,然后在后台对 js, css 进行合并,这样对于浏览器来说依然是一个请求,但是开发时仍然能还原成多个,方便管理和重复引用。yahoo 甚至建议将首页的 css 和 js 直接写在页面文件里面,而不是外部引用。

因为首页的访问量太大了,这么做也可以减少两个请求数。而事实上国内的很多门户都是这么做的。

而 css sprites 是指只用将页面上的背景图合并成一张,然后通过 css 的 background-position 属性定义不过的值来取他的背景。

## Yahoo 前端优化十四条军规 II

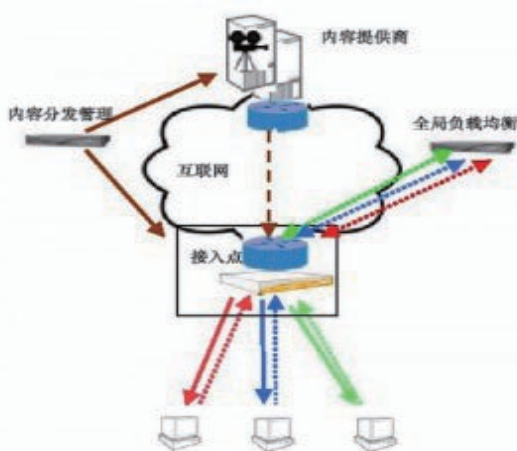
淘宝和阿里巴巴中文站目前都是这样做的。  
有兴趣的可以看下淘宝和阿里巴巴的背景图。

<http://www.csssprites.com/> 这是个工具网站，  
它可以自动将你上传的图片合并并给出对应的  
background-position 坐标。并将结果以 png 和 gif  
的格式输出。

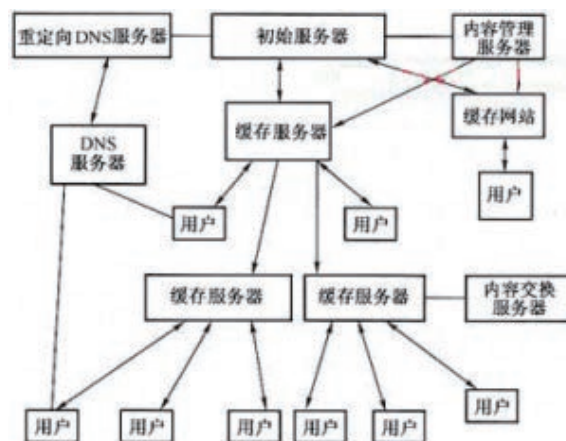
### 第二条、使用 CDN (内容分发网络): Use a Content Delivery Network

说实话,对于 CDN 这一块自己并不是很了解,简单地讲,通过在现有的 Internet 中增加一层新的网络架构,将网站的内容发布到最接近用户的 cache 服务器内,通过 DNS 负载均衡的技术,判断用户来源就近访问 cache 服务器取得所需的内容,杭州的用户访问近杭州服务器上的内容,北京的访问 近北京服务器上的内容。这样可以有效减少数据在网络上传输的时间,提高速度。更详细地内容大家可以参考百度百科上对于 CDN 的解释。Yahoo! 把静态内容分布到 CDN 减少了用户影响时间 20% 或更多。

CDN 技术示意图:



CDN 组网示意图:



### 第三条、添加 Expire/Cache-Control 头: Add an Expires Header

现在越来越多的图片,脚本,css,flash 被嵌入到页面中,当我们访问他们的时候势必会做许多次的 http 请求。其实我们可以通过设置 Expires header 来缓存这些文件。

Expire 其实就是通过 header 报文来指定特定类型的文件在浏览器中的缓存时间。大多数的图片,flash 在发布后都是不需要经常修改的,做了缓存以后这样浏览器以后就不需要再从服务器下载这些文件而是而直接从缓存中读取,这样再次访问页面的速度会大大加快。

### 第四条、启用 Gzip 压缩: Gzip Components

Gzip 的思想就是把文件先在服务器端进行压缩,然后再传输。这样可以显著减少文件传输的大小。

传输完毕后浏览器会 重新对压缩过的内容进行解压缩,并执行。

目前的浏览器都能“良好”地支持 gzip。不仅浏览器可以识别,而且各大“爬虫”也同样可以识别,这样各位 seoer 可以放心了。



## Yahoo 前端优化十四条军规 III

而且 gzip 的压缩比例非常大,一般压缩率为 85%,就是说服务器端 100K 的页面可以压缩到 25K 左右再发送到客户端。具体的 Gzip 压缩原理大家可以参考 csdn 上的《gzip 压缩算法》这篇文章。雅虎特别强调,所有的文本内容都应该被 gzip 压缩:html (php), js, css, xml, txt... 这一点我们网站做得不错,是一个 A。以前我们的首页也并不是 A,因为首页上还有很多广告代码投放的 js,这些广告代码拥有者的网站的 js 没有经过 gzip 压缩,也会拖累我们网站。

以上三点大多属于服务器端的内容,本人也是粗浅地了解而已。说得不对的地方有待各位指正。

### 第五条、将 css 放在页面最上面 ( Put Stylesheets at the Top )

将 css 放在页面最上面,这是为什么? 因为 ie, firefox 等浏览器在 css 全部传输完全之前不会去渲染任何东西。

理由诚如小马哥说得那样很简单。css,全称 Cascading Style Sheets (层叠样式表单)。层叠即意味这后面的 css 可以覆盖前面的 css,级别高的 css 可以覆盖级别低的 css。在 [css 之! important] 这篇文章的最下面曾简单地提到过这层级关系,这里我们只需要知道 css 可以被覆盖的。既然前面的可以被覆盖,浏览器在他完全加载完毕之后再去渲染无疑也是合情合理的很多浏览器下,如 IE,把样式表放在页面的底部的问题在于它禁止了网页内容的顺序显示。浏览器阻止显示以免重画页面元素,那用户只能看到空白页了。

Firefox 不会阻止显示,但这意味着当样式表下载后,有些页面元素可能需要重画,这导致闪烁问题。所以我们应该尽快让 css 加载完毕

顺着这层意思,如果我们再细究的话,其实还有可以优化的地方。比如本站上面包含的两个 css 文件,

### 第六条、将 script 放在页面最下面 ( Put Scripts at the Bottom )

将脚本放在页面最下面的目的有那么两点:

1、因为防止 script 脚本的执行阻塞页面的下载。在页面 loading 的过程中,当浏览器读到 js 执行语句的时候一定会把它全部解释完毕后在会接下来读下面的内容。

不信你可以写一个 js 死循环看看页面下面的东西还会不会出来。( setTimeout 和 setInterval 的执行有点类似于多线程,在相应的响应时间之前也会继续下面的内容渲染。)浏览器这么做的逻辑是因为 js 随时可能执行 location.href 或是其他可能完全中断此页面过程的函数,即如此,当然得等他执行完毕之后再加载咯。所以放在页面最后,可以有效的减少页面可视元素的加载时间。

2、脚本引起的第二个问题是它阻塞并行下载数量。HTTP/1.1 规范建议浏览器每个主机的并行下载数不超过 2 个(IE 只能为 2 个,其他浏览器如 ff 等都是默认设置为 2 个,不过新出的 ie8 可以达 6 个)。因此如果您把图像文件分布到多台机器的话,您可以达到超过 2 个的并行下载。

■未完,查看网络原文:

<http://developer.51cto.com/art/201207/347525.htm>

# 我在谷歌的一次有趣的面试经历

作者 / 穿山甲

很多年前我进入硅谷人才市场,当时是想找一份高级工程师的职位。

如果你有一段时间没有面试过,根据经验,有个非常有用的提醒你应该接受,就是:你往往会在前几次面试中的什么地方犯一些错误。简单而言就是,不要首先去你梦想的公司里面试。

面试中有多如牛毛的应该注意的问题,你可能全部忘记了,所以,先去几个不太重要的公司里面试,它们会在这些方面对你起教育(再教育)作用。

我第一家面试的公司叫做 `gofish.com`,据我所知, `gofish` 这家公司如今的情况跟我当时面试时完全的不同。我几乎能打保票的说,当时我在那遇到的那些人都已不再那工作了,这家公司的实际情况跟我们这个故事并不是很相关。但在其中的面试却是十分相关的。对我进行技术性面试的人是一个叫做 `Guy` 的家伙。

`Guy` 穿了一条皮裤子。众所周知,穿皮裤子的面试官通常是让人“格外”恐怖的。而 `Guy` 也没有任何让人失望的意思。他同样也是一个技术难题终结者。而且是一个穿皮裤子的技术难题终结者——真的,我做不到他那样。

我永远不会忘记他问我的一个问题。事实上,这个问题是非常的普通——在当时也是硅谷里标准的面试题。

问题是这样的:假设这有一个各种字母组成的字符串,假设这还有另外一个字符串,而且这个字符串里的字母数相对少一些。从算法是讲,什

么方法能最快的查出所有小字符串里的字母在大字符串里都有?

比如,如果是下面两个字符串:

String 1: ABCDEFGHLMNOPQRS

String 2: DCGSRQPOM

答案是 `true`,所有在 `string2` 里的字母 `string1` 也都有。如果是下面两个字符串:

String 1: ABCDEFGHLMNOPQRS

String 2: DCGSRQPOZ

答案是 `false`,因为第二个字符串里的 `Z` 字母不在第一个字符串里。

当他问题这个问题时,不夸张的说,我几乎要脱口而出。事实上,对这个问题我很有信心。(提示:我提供的答案对他来说显然是最糟糕的一种,从面试中他大量的各种细微表现中看得出来)。

对于这种操作一种幼稚的做法是轮询第二个字符串里的每个字母,看它是否同在第一个字符串里。

从算法上讲,这需要  $O(n \cdot m)$  次操作,其中  $n$  是 `string1` 的长度,  $m$  是 `string2` 的长度。就拿上面的例子来说,最坏的情况下将会有  $16 \cdot 8 = 128$  次操作。

一个稍微好一点的方案是先对这两个字符串的字母进行排序,然后同时对两个字串依次轮询。两个字串的排序需要(常规情况)  $O(m \log m) + O(n \log n)$  次操作,之后的线性扫描需要  $O(m+n)$  次操作。

## 我在谷歌的一次有趣的面试经历 II

同样拿上面的字串做例子,将会需要  $16 \times 4 + 8 \times 3 = 88$  加上对两个字串线性扫描的  $16 + 8 = 24$  的操作。(随着字串长度的增长,你会发现这个算法的效果会越来越好)

最终,我告诉了他一个最佳的算法,只需要  $O(n+m)$  次操作。方法就是,对第一个字串进行轮询,把其中的每个字母都放入一个 Hashtable 里 (成本是  $O(n)$  或 16 次操作)。然后轮询第二个字串,在 Hashtable 里查询每个字母,看能否找到。如果找不到,说明没有匹配成功。这将消耗掉 8 次操作 —— 这样两项操作加起来一共只有 24 次。不错吧,比前面两种方案都要好。

Guy 没有被打动。他把他的皮裤子弄的沙声响作为回应。“还有没有更好的?”他问道。

我的天? 这个家伙究竟想要什么? 我看看白板,然后转向他。“没有了,  $O(n+m)$  是你能得到的最好的结果了 —— 我是说,你至少要对每个字母至少访问一次才能完成这项操作 —— 而这个方案是刚好是对每个字母只访问一次。”我越想越确信我是对的。

他走到白板前,“如果这样呢 —— 假设我们有一个一定个数的字母组成字串 —— 我给每个字母分配一个素数,从 2 开始,往后类推。这样 A 将会是 2, B 将会是 3, C 将会是 5, 等等。现在我遍历第一个字串,把每个字母代表的素数相乘。你最终会得到一个很大的整数,对吧? 然后 —— 轮询第二个字符串,用每个字母除它。如果除的结果有余数,这说明有不匹配的字母。如果整个过程中没有余数,你应该知道它是第一个字串恰好的子集了。这样不行吗?”

每当这个时候 —— 当某个人的奇思异想超出了你的思维模式时,你真的需要一段时间来跟上他的思路。现在他站在那里,他的皮裤子并没有帮助我理解他。

现在我想告诉你 —— Guy 的方案 (不消说,我并不认为 Guy 是第一个想出这招的人) 在算法上并不能说就比我的好。而且在实际操作中,你很可能仍会使用我的方案,因为它更通用,无需跟麻烦的大型数字打交道。但从巧妙水平上讲, Guy 提供的是一种更、更、更有趣的方案。

我没有得到这份职位。也许是因为我拒绝了他们提供给我的一些讨厌的工作内容和其它一些东西,但这都无所谓了。我还有更大更好的目标呢。接着,我应聘了 become.com。在跟 CTO 的电话面试中,他给我布置了一道“编程作业”。这个作业有点荒唐 —— 现在回想起来,大概用了我 3 天的时间去完成。我得到了面试,得到了那份工作 —— 但对于我来说,最大的收获是这道编程作业强迫我去钻研并有所获。我需要去开发一个网页爬虫,一个拼写检查 / 纠正器,还有一些其它的功能。不错的东西。然而,最终,我拒绝了这份工作。



终于,我来到了 Google 面试■未完,查看网络原文:

<http://developer.51cto.com/art/201207/347641.htm>



# King.com产品经理谈HTML5发展

作为一种用于创造网页游戏的新工具,HTML5已经成为了2012年社交游戏领域的一大热词。King.com最近发布于Facebook的《Pyramid Solitaire Saga》便是一款基于HTML5技术的游戏



作为一种用于创造网页游戏的新工具,HTML5已经成为了2012年社交游戏领域的一大热词。King.com最近发布于Facebook的《Pyramid Solitaire Saga》便是一款基于HTML5技术的游戏(游戏邦注:但德国社交游戏开发商Wooga则在最近宣布放弃HTML5),该公司产品经理Levina Nilsson在最近媒体采访中解释了King.com看好HTML5技术的原因。



levina-nilsson(from games)

**为什么你们决定使用HTML5创造《Pyramid Solitaire Saga》?**

虽然原因有许多,但主要是为了跨平台为玩家创造出无缝的游戏体验,并希望通过《Pyramid Solitaire Saga》去测试我们在这方面的技术水平。我们希望玩家能够根据自己的需求而随意选择在台式电脑,平板电脑或智能手机上玩同一款游戏,并且无需下载原生应用。我们同样也希望减少

电脑与手机开发间的障碍,所以我们才决定尝试HTML5并观察它是否是一个可行的游戏平台。

**比起去年出现的HTML5游戏,《Pyramid Solitaire Saga》不管是从外表还是音效看来都更加吸引人,而你们又是如何做到这一点呢?**

作为一家领先的休闲社交游戏开发商,我们始终坚持站在跨平台创新与开发策略的前沿。而为了实现这一目标,King.com已经投入了大量的资源去吸引更多优秀开发团队。我们不只是想要尝试HTML5,我们也想要创造出一款非常出色的游戏,在之前成就的基础上向前再进一步。因此我们在游戏中添加了许多图像资源,并不断完善游戏理念,希望将其打造成Facebook上最受欢迎的纸牌类游戏——《Pyramid Solitaire Saga》在发布后一个月内便获得了超过100万的日活跃用户。

当然我们也未放弃其它游戏。我们拥有一支团队专门致力于维护游戏的持续运行,并为玩家创造出更棒的游戏体验。今天,我们不仅拥有Facebook上最受欢迎的纸牌游戏,从总体的月活跃用户来看,King.com也是排名第二的Facebook游戏开发商。

**HTML5之前一直难以成为众开发者所期待的领先游戏开发工具,那么现在的它是否已经“修成正果”了?而这对于玩家来说又有何意义?**

## King.com 产品经理谈 HTML5 技术发展潜力 II

答案是还没有。尽管 HTML5 已经成为了众人的关注焦点,但是它仍然是一种非常年轻的技术,也只有那些足够成熟的开发者才能驾驭得了它。

除此之外我们也还未能创造出最佳游戏体验。但这不仅是因为 HTML5 还未准备好,市场上大多数手机设备和浏览器也还未准备好迎接这一新技术的降临。所以现在的玩家仍需要继续下载手机游戏的原生应用,并且他们所经历的游戏体验也还不甚完美。



使用 Flash 和 Unity 等工具我们已能够创造出华丽的 3D 图像和让人震撼的特效,而你是否也希望能够在 HTML5 中看到这些功能? 这是否是 HTML5 游戏开发者所关心的内容?

我想在早期阶段我们最好不要在这一新技术身上抱有太多期望。根据我们的经验,如果你需要某些内容,你就必须亲自参与创造这些内容;因此我们便与谷歌保持紧密合作并致力于研究他们的开放源码解决方案 PlayN。

同时为了更好地进行游戏开发,我们还创造了自己的游戏框架 Wrath。Wrath 与 PlayN 的结合有助于我们创造出一款单独的游戏并将代码基应用于所有设备上(注:这时候便不再需要面向网页去使用 Flash 或者面向 iOS 去使用 C++)。可以说 Wrath 是出自 King.com 之手的第一个开放资源。

而我们也希望有更多开发者加入这个探索过程,努力推动 HTML5 成为一种真正有效的游戏开发模型。

今后 King.com 是否还会推出更多 HTML5 游戏?

会,尽管 HTML5 现在还存在着种种不足,但是我们仍然相信这是一种极具潜力的工具。

现在我们已经知道为何开发者们会转向 HTML5 了,但是我们却不清楚它能满足玩家的何种需求?

玩家终究希望能够在平板电脑,智能手机以及电脑上享受到一种完全同步的游戏体验,而这也正是 HTML5 的长处所在。玩家们只是还未注意到 HTML5 的优势。HTML5 正如人们预期的那样能够兼容多个平台。

如果玩家有这种需求,那么即使是创造智能手机的大型公司也将不得不研发能够处理 HTML5 技术的产品。

对于我们的用户来说时间和便利便是最宝贵的财富,而在不远的未来 HTML5 (它可让用户可以无需下载原生应用或在不同设备上追踪游戏情况)便能够帮助他们有效地储存这些财富。■

# 非小型电子商务系统设计经验分享

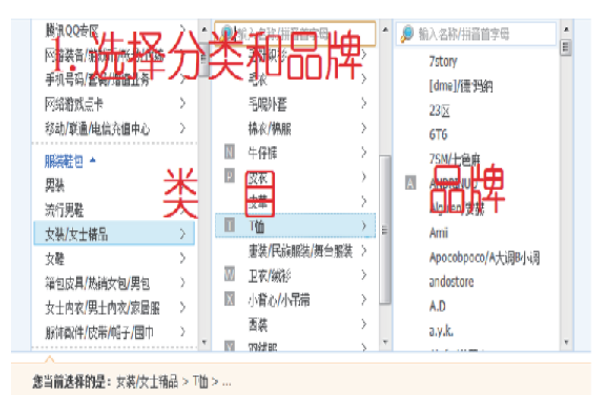
## 前言

做了两年多针对淘宝的电子商务数据线下数据系统,越到后面越觉得自己还没入门,不管技术上还是业务上,这篇文章既是对自己的积累的一次梳理,更想的是能在和各位朋友交流中,互相进步。

ps: 所有字段并不是正式项目所使用字段,请根据自己的业务需求进行酌情查看处理,类目属性,商品,订单结构可以参考淘宝 API 数据接口进行查看具体字段。

## 商品模块设计

商品模块是支撑整个架构的核心,如果这块没设计好,那么所有后期的复杂的统计需求基本都满足不了。为什么这样子设计属性看这里和这里(注:所有这里的超链接请去原文中查看),把品牌从类目中剥离出来是为了降低程序针对商品属性这块的复杂度。这里通过淘宝的添加宝贝的操作来说明上面的数据结构如何满足需求。



PS: 本来要截玉兰油沐浴露的图,结果发现淘宝取消了以前选择毫升 \* 买的多送得多组合

SKU 的添加商品方式,改成了一个 SKU 就是一个宝贝的编辑手段,呵呵,没办法,只有上面截个衣服的图,下面的数据却是快消品的。淘宝这样做这也是没办法的,这种快消品不同 SKU,图片还能用一样的,而且大部分用户搜索的时候呢,会喜欢直接搜索具体的毫升数,这也给我们提了个醒,不同的类目可能会是不一样的处理方式,就算是服装这种 SKU 相对标准的类目,也会有说在展示和搜索结果中,会放置一个产品的多个 SKU,比如凡客的网站,一件衣服的几个颜色都会出现在类目搜索结果中,增加曝光度,吸引用户点击购买。页面属性的编程实现可以参考这里。SKU 存放在产品 SKU 表中,按我们的实际需求增加修改字段,比如我的表中多了 ProductCode 和 BarCode 字段,SKU 的属性会拆分后存入产品基本属性值表,便于搜索或统计等需求。

这里我是把所有的属性都打成一条一条存储在这个表中(注:图表请进原文中看),那么能满足我们在日常业务的属性搜索,统计等需求。按属性搜索,这里必须要注意以下几点:

1. 不可能所有的属性都开放给用户或者我们的客户进行搜索,所以我们会在属性名表中有个字段(是否搜索字段)来人工控制哪些属性是搜索属性

2. 基本属性是同一个宝贝下面所有 SKU 都共有的,SKU 属性是单个 SKU 独有的,所以搜索的时候还必须分清楚销售属性(销售属性组成 SKU)和基本属性。



## 非小型电子商务系统设计经验分享 II

3. 属性图片的存储我并没有设计,因为我们 是做快消品,没有这个需求。但是,如果我做的 话我还会是在基本属性值表中加上“是否图片属性, 是否使用默认图片,图片 URL “3 个字段来记录 颜色属性。做属性搜索的时候比较方便。

4. 产品通过关键字搜索和属性搜索是分开的,两种搜索并不是一种解决方式,比如淘宝,在 首页的搜索框是通过分词匹配宝贝标题的关键 字,通过关键字的匹配程度,店铺的 dsr 评分权重 来决定搜索结果,而属性搜索的时候则是匹配满 足属性条件的宝贝。那属性又分第 1 点和第 2 点, 所以还是挺麻烦的。

那到了这里产品的存储已经说完了,其它的 运费什么的,就懒得说了。

这里你会发现有打包品表,打包品子表,最 终商品表,商品变更记录表。这里需要详细说明 一下。

### 首先说一下打包品概念:

打包品: 为了各种运营上的需求,很多时候 我们会人为的把多个 SKU 组合成一个商品进行 组合销售,我们在淘宝购物的时候,经常会看到 这样的情况, A 产品 +B 产品组合销售, AB 的组合 在淘宝上面表现为一个宝贝,你看看这里或者 这里或者这里,这些就是拉。这种销售数据在订 单里表现是一个淘宝商品,但是我们要做库存管理, 数据分析等需要拆分出来。这是必须考虑的!

PS: 有那种出厂打包品,比如一个包装盒 里面有香皂,有沐浴露,但是它们本身就是一个 SKU,出厂就这样,所以不能和打包品混为一谈。

由于我们运营上的需要,我们可能卖单个 SKU,也可能卖多个 SKU 的组合,那么在我这里 把单个 SKU 和多个 SKU 组合都看成打包品,单 个的 SKU 打包品它的子项只有它自己,这样做 的好处是,系统中只需要一种方式来处理这种关系。 在打包品表中通过类型来区分。

这里有一个关键问题要注意,我们在出售商 品过程中,价格是可能会随时人工或者系统来干 预变化的,比如产品 A 标题叫 B 洗发水 +C 护发 素直降 20 元,但是我们根据实际的流量和转换率 价格可以上下浮动,那么我们就及时的调整价 格,所以我们的标题,价格都需要进行更改,这里 牵涉 2 个问题,我们是新建一个打包品或者我们 是另外放在最终商品表,我们就需要修改对应的 标题和价格,同时呢,在商品变更记录表中记录添 加一个上次修改的备份,作为我们对不同价格的 转换率的一个分析基础数据。

第二个问题就是由于修改了打包品或者创 建了新的打包品(SKU 子项, SKU 数量一样)价 格,那么对应的分配到每个具体 SKU 的价格发生 了变化,这里如果是新建了打包品就没问题,但 是如果是修改打包品,那么我们对打包品 SKU 子 项的价格就必须通过相应的公式进行计算。比 如 A+B+C 今天是 100 元, A 是 30, B 是 50, C 是 20, 如果价格变成了 90 或者 110,那么对应到具 体的子项价格也需要更改,因为很经常的需求就 是统计某产品或者某 SKU 的销售量和销售额。

■未完,查看网络原文:

<http://developer.51cto.com/art/201207/346934.htm>



# 腾讯发布《Web App白皮书》

白皮书中,腾讯一方对自家的 QQ 浏览器的功能进行了“大力度”的推荐,其主要侧重于 HTML5 功能如何牛掰。不过在该白皮书中,gamelook 也注意到一些体现腾讯移动 web 业务特色的说法。

GameLook 报道 / 6 月 27 日消息,手机 QQ 浏览器 HTML 5 创新大赛沙龙分享会第二场在北京成功举行。会上腾讯发布了《手机 QQ 浏览器与 Web App 开发者共赢白皮书》(以下简称《Web App 白皮书》),详细分析了 Web App 的未来发展趋势,HTML5 技术前景,以及腾讯在推动 HTML5 技术进步,为 Web App 开发者提供优秀研发环境上所做的努力。

白皮书中,腾讯一方对自家的 QQ 浏览器的功能进行了“大力度”的推荐,其主要侧重于 HTML5 功能如何牛掰。不过在该白皮书中,gamelook 也注意到一些体现腾讯移动 web 业务特色的说法:

**1 支付方面,腾讯 QQ 浏览器提供了一些非腾讯支付渠道。**

支持短信、Q 币、手机充值卡、QQ 卡、财付通、银行卡、易贝等支付手段,建立了一个覆盖广泛,支付方式全面的支付体系。

## 2 用户行为数据分析

QQ Mobile Browser User's Profile 定向、推荐系统,为开发者提供个性化服务、精细运营的支撑系统;作为基础能力,提高浏览器商业价值;为分领域、多维度构建用户兴趣模型(画像)提供基础数据;尝试构建、挖掘基于用户兴趣模型的推荐系统(引擎);清晰、灵活的 Kernel-Service 架

构模式, app-oriented;

## 3 补充手段,插件体系

鉴于目前 HTML5 标准仍未正式制定,技术有待继续发展和成熟,因此插件是手机浏览器拓展其应用的有效手段。目前手机 QQ 浏览器(Android) 主线功能已趋于完善,一些需求及功能已属于传统浏览器的插件及附加组件,目前手机 QQ 浏览器主线版本已有深度集成的类似插件的功能,包括“应用中心”、“FLASH 播放”、“支付宝及财付通”等。



实现 W3C 标准中定义的设备相关的操控能力,包括电池、摄像头、地理信息、感应器等,此外还支持震动、通知、网页文件读写等 Device API 标准,拥有对 Device API 的优秀支持能力,可以完美支持 Web APP 游戏、看视频、拍照等功能,手机端 Web APP 后续的开发和实现奠定了技术基础。

■未完,查看网络原文:

<http://developer.51cto.com/art/201206/345302.htm>

# 程序员怎么提高英语阅读水平

今天在知乎上被邀请回答这个问题,一不小心写了很多,算算已经可以当作一篇博客了,于是记录下来。

**问题:** 作为一名程序员,虽说每天都在和英语打交道,但是当看到一篇英语文档或者英语技术文章的时候还是比较头疼,理解他们的意思也只能是 20%。尤其是使用 google 搜索的时候,很多问题解决办法都是英文的,还有一些国外比较有名的网站比如 stack overflow,上面也有很多学习的资源。怎样才能让自己顺利阅读这些技术文章呢?

**回复:** 其实学英语和其他技能没什么太大区别,无论是你想在英语阅读、口语或写作方面提升自己,都需要进行长期的不间断的练习,坚持一段时间后(时间长短根据你的效率、每天用时、频率都有关系),你会发现自己的水平自然就提升了。举个例子,现在写博客似乎是最难坚持的,但是如果你每个月都能写一篇略有价值文章的话,5 年就会有 60 篇高质量的博客,你几乎都能集结出书了。最简单的坚持,最后的结果都是惊人的。但很多人往往每年只能写 1 篇,或者干脆一放就几年过去了。所以你知道,成事的人总是少数的。作为程序员,英语阅读能力是最基本的要求,相对口语和写作来说也是最容易达到的,因为计算机类图书的那些常用单词就那么多,多读几本英语类技术图书,想不认识都难。

一个相对容易坚持的办法就是,找一本和当前工作相关的、急需的技术图书,每天拿出一小时阅读,不认识的单词,如果不影响阅读可以不查词

典。如果某个单词多次出现,那么就该查下词典并计入生词本。如果你能坚持查阅英英词典,那么提高就会更快了。

每天坚持一小时,这本书读完,你就会发现自己的阅读能力提高了一大截。这样看来提高阅读能力似乎很简单,但是,问题的关键是,你能否坚持下来,尤其是阅读初期,你只能看懂 20%,需要不停的查字典,单词记了又忘,几个星期过去了,进展依然缓慢。这时候最容易放弃,你会痛苦的怀疑自己,我究竟是不是这块料。毕竟人类的大脑都是倾向于舒适和懒惰的,谁都知道花一个或几个小时看看美剧或刷刷微薄,是容易的,嗨皮的,时间在感知上也是,刷一下就过去了。学英语、学一门新的编程语言、写博客是痛苦的,有时候你甚至会有意识的去避免开始这件事情,先干点这个,再干点那个,熬到最后,发现没时间了,再拿起书来读一会,困了,今天先睡吧。就这样,一天天很快过去了,你发现自己似乎每天都在坚持,但依然没效果。但事实是,你既没坚持,也没效率。

所以,提高英语阅读能力这事,和学其他的技能区别不大,而且是相对容易的,比提高英语口语和写作能力容易多了。如果程序员缺乏英语阅读能力,这将是您非常大的一块短板,如果意识到了,尽早补上。

另外说一点,英语翻译对提升英语阅读有一定的帮助,但这个要求比较高,必须要查字典,遣词造句,力求原汁原味的表达作者的意图。并不是提升阅读能力的捷径。最后说一句,10000 小时的练习时间,真不是盖的! ■

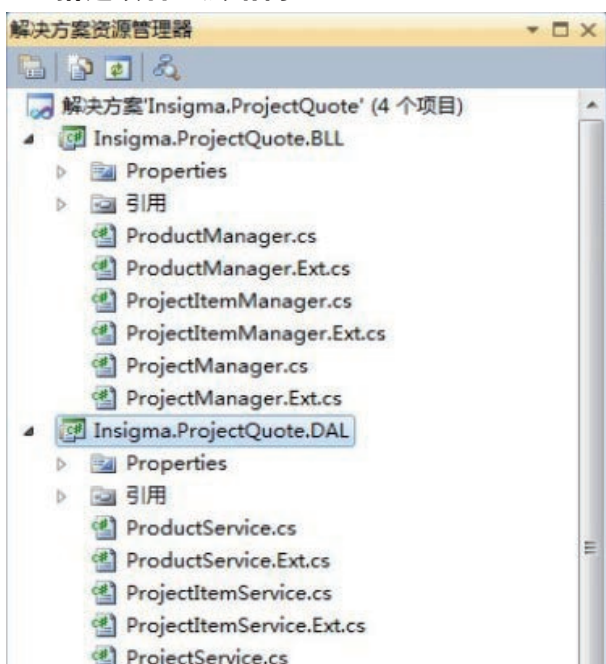
## ■ 编者按

这次来讲 MVC 中最后的 M。Model, 几乎可以说是网页应用的核心。之前课程提到过网页应用是由数据库驱动, 而在很多场景, 数据库 = M ; M = 数据库。

# 两天完成一个小型工程报价系统

花了两天, 整理了一下三层架构, 顺便练了一个小型三层架构——工程报价系统。功能很简单, 完成基本的增删改查。

## 搭建项目三层结构



## 界面的设计以及美化



## 分层代码的设计与实现

1. 模型层: 由于表设计时存在主外键关系, 故把表映射成对象时添加一个外键对象来保存外键那张表的相关数据。

// 外键实体

```
private Product _product;

private Project _project;

public Project _Project {
    get { return _project; }
    set { _project = value; }
}

public Product _Product {
    get { return _product; }
    set { _product = value; }
}
```

2. 数据访问层: 把数据封装成 Model 对象的各层之间传递。

我个人不喜欢三层之间传递 DataSet 数据集, 我的理解是尽量不在 BLL 层出现 DataSet, 所以我在 DAL 层把所有获取的数据封装成 IEnumerable 集合, 然后返回。

```
protected IEnumerable<ProjectItem> ToModelByFK(SqlDataReader reader)
{
    var list = new List<ProjectItem>();
    while (reader.Read()) {
        list.Add(ToModelByFK(reader));.....
    }
}
```

■ 本文未完, 详细部分请参考原文

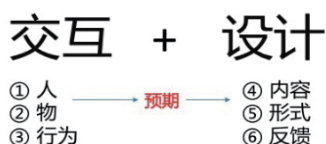
<http://developer.51cto.com/art/201206/343329.htm>

# 交互设计那些事儿：开发人员必备

这是一次内部分享，新部门的同学们希望我聊聊我过去从事的交互设计，做个了面对入门者等级的 PPT，和各位简单聊了聊。发到微博上，也敦促自己对交互设计从业经历做个简单总结。——类似的文章，貌似自己在三年前也写过……

## 交互设计是什么？

以**人的需求**为导向，理解**用户**的期望、需求，理解**商业、技术**以及业内的机会与制约。基于以上的理解，**创造出形式、内容、行为**有用，易用，令用户满意且技术可行，具有商业利益的产品。



### 一．什么是交互设计

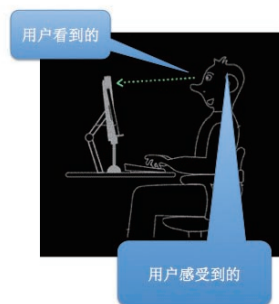
要谈什么是交互设计，先要了解这几个概念。

网页设计=？

美工=？

UI=？

UED=？



网页设计年代，大概是从 2004 年开始热起来了，等到我毕业的 2006 年，满大街都已经是电脑培训机构，记得当时北大青鸟的百日千才培训就在我每日下班必经的十字路口。

网页设计年代重视的是特效，说白了就是特效背后的技术，君不见当时的网页是让人看得眼花缭乱啊，这里 bling bling 闪闪的，漂浮框乱跳的。然后就是套各种网页设计模版，做出很多复杂的

类似于右边这种典型的政府网站出来。

### 美工时代：

网页设计时代，大批的“美工”涌现。显然人们有了更高的追求，光是 bling bling 的眼花缭乱的动画效果已经看厌了，网页也是需要装饰美化的。虽然网页设计论坛(www.68design.net)当时很火，但是网页设计师这个头衔仍然是一厢情愿的叫法。大多数时候，仍然是老板和客户嘴里的“美工”。

美工时代重视的是：视觉效果。并且，原来对网页设计一窍不通的老板们纷纷成了资深的视觉鉴赏家。美工时代“美工”很痛苦——单纯靠视觉效果去评价作品好坏，太主观，太随意。于是他们通常会做 N 个版本，反复修改，直至美工崩溃或者老板崩溃。

### 历史迈入了伟大的 UI 年代。

大概是在 2005 年，我崇拜的网络名人从擅长鼠绘的小非变成了图标达人 Rokey（时光倒流到 2005 年，我正在看这篇文章：灵感来源于生活——微软亚洲工程院移动设备组 UI 设计师张伟）。也自学了关于 UI 的若干知识，开始迷恋起 PS 像素级的奇妙世界。

UI=User Interface. 已经将 User 即用户纳入界面设计考虑的重要要素。但是对于精致、质感



## 交互设计那些事儿：开发人员必备 II

的无极限的追求,依然会让人顾此失彼,忘记了从用户的视角去审视一个 UI 的作品。于是,经常会看到虽然很精致,但是很难用的软件界面。

用户看到的部份,无可挑剔。但是实际上,使用的感受并不好。而感受部份,可以从几个角度去阐释:

他们知道不知道这个是什么东西? 他们感觉到迷惑吗?

他们是否清楚能够通过此物做什么?

是否清楚如何做到?

◆ 当他们有兴趣去尝试时,他们知道如何开始吗?

◆ 当他们开始后,知道如何进行下一步吗?

◆ 当不想要使用时,能够快速明白如何退出吗?

◆ 当他们退出后,下次来用,是否需要重新去学习掌握如何操作?

◆ 每一步操作时,他们心理会感觉到愉悦还是枯燥乏味?

◆ 当然,我们崇尚软件操作过程中的效率,但是若用户感觉到愉悦,也许就不会觉得效率太低,其实,效率是个相对的说法。归根结底,这就是用户体验。

“UI”时代



视觉震撼,每日没夜  
精致,质感,质感!



## UED 时代——用户体验制胜

我们不缺技术,我们不缺另人震撼的视觉,这些虽然过去是我们孜孜追求的东西,如今已经不足为奇。但是我们缺用户来用,我们缺用户持续来关注,我们缺用户体验。用户体验过去一直在产生影响,但是未能得到足够的重视。

2007 年,开始关注 [www.ucdchina.com](http://www.ucdchina.com). UED 是套方法论,UCD 是指导思想——User-centered Design. 不再是 BCD-Boss-Centered Design. 老板说了算,除非老板能够代表目标客户。这和市场营销学发展的历程有点像,从生产主导到市场主导。

### UED 用户体验设计



以下是我在用户体验设计? ——是什么不是什么里用到的图片。WEB2.0 后,用户的参与度越来越高,用户从被动的接受者变成了主笔,主编,摄影师,作家……他们是主角,使用中遇到问题,当然不可能靠网站的客服电话或帮助教程搞定,他们需要自助,所以,降低使用门槛,消除使用障碍,吸引他们重复使用,是每个网站致力的方向。



## 交互设计那些事儿：开发人员必备 III

以下是在用户体验设计答疑对话(半吊子和纯外行)一文中的配图。从广义来讲,一个网站的任何部门都是在为用户体验服务,从战略层的布局开始,用户体验已经开始。而狭义的用户体验设计,也就是 UED 部门做的事情,大部分是围绕到结构、框架以及表面层。

UED 部门的组织架构基本如下,解决用户易用(交互、文案),以及想用(视觉、交互)的问题。而用户研究是帮我们看清用户特征、洞察及挖掘用户需求,前端是伟大的实现者。

文案对于用户体验的重要性经常被忽视,所以,文案撰写在 UED 团队是很稀缺的资源。目前据我了解,只有支付宝的 UED 团队有此配置,其他团队大部分是由产品经理担任此职。

### UED架构保障

- 视觉设计—Visual Design
- 交互设计—Interaction Design
- 用户研究—User Research
- 前端开发—Web Develop Engineer
- 文案撰写—Copy Writer

那么,交互设计到底是什么呢?

我们不缺技术,我们不缺另人震撼的视觉,这些虽然过去是我们孜孜追求的东西,如今已经不足为奇。但是我们缺用户来用,我们缺用户持续来关注,我们缺用户体验。用户体验过去一直在产生影响,但是未能得到足够的重视。

我突然打了一下你,你肯定会有一定的反应。首先,你的表现会吃惊或者生气,你的吃惊或者生气的程度也会受我下手的轻重和方式影响。

我呢,对你的反应会有一定的预期,下手的时候,没有期望你会突然哈哈大笑——这就是人

与人之间的交互。

人和物也是一样,我按下电视的开关键,我预期电视会打开,并且电视如期望发生运转。若我按下电视的开关键,突然电视成了静音,这就是非期望的交互行为。

### 二. 为什么需要交互设计?

人和物也是一样,我按下电视的开关键,我预期电视会打开,并且电视如期望发生运转。若我按下电视的开关键,突然电视成了静音,这就是非期望的交互行为。

人可以经由训练,改变自己对于行为的反馈,比如,妈妈教育婴儿对于他人的礼物回报以“谢谢”和微笑,不要跟着陌生叔叔走等。如果你送一个小孩子礼物,他却对于大骂,你会觉得这个小子怎么这么没教养。

机器、系统没有生命力,则需要被赋予对于各种行为的反馈机制。将用户的期望赋予给它,让它给出合理的反馈行为。

然而不幸的是,我们的生活里,没有教养的产品实在太多太多……

#### 不好的交互



上图左侧是一个杯子。我让现场的同学猜这个杯子的成本价,答案集中在 10 元左右。但是不幸的是,我为这个杯子花费了不下 600 元。原因正在于不良交互。

■ 本文未完,详细部分请参考原文

<http://developer.51cto.com/art/201206/345028.htm>

# Python语言十分钟快速入门

Python（蟒蛇）是一种动态解释型的编程语言。Python 可以在 Windows、UNIX、MAC 等多种操作系统上使用,也可以在 Java、.NET 开发平台上使用。

## 【简介】

Python（蟒蛇）是一种动态解释型的编程语言。Python 可以在 Windows、UNIX、MAC 等多种操作系统上使用,也可以在 Java、.NET 开发平台上使用。



## 【特点】

1 Python 使用 C 语言开发,但是 Python 不再有 C 语言中的指针等复杂的数据类型。

2 Python 具有很强的面向对象特性,而且简化了面向对象的实现。它消除了保护类型、抽象类、接口等面向对象的元素。

3 Python 代码块使用空格或制表符缩进的方式分隔代码。

4 Python 仅有 31 个保留字,而且没有分号、begin、end 等标记。

5 Python 是强类型语言,变量创建后会对应一种数据类型,出现在统一表达式中的不同类型的变量需要做类型转换。

## 【搭建开发环境】

1 可以到 [www.python.org](http://www.python.org) 下载安装包,然后通过 configure、make、make install 进行安装。

2 也可以到 [www.activestate.com](http://www.activestate.com) 去下载 ActivePython 组件包。(ActivePython 是对 Python 核心和常用模块的二进制包装,它是 ActiveState 公司发布的 Python 开发环境。ActivePython 使得 Python 的安装更加容易,并且可以应用在各种操作系统上。ActivePython 包含了一些常用的 Python 扩展,以及 Windows 环境的编程接口)。对 ActivePython 来说,如果你是 windows 用户,下载 msi 包安装即可;如果你是 Unix 用户,下载 tar.gz 包直接解压即可。

3 Python 的 IDE, 包括 PythonWin、Eclipse+PyDev 插件、Komodo、EditPlus

## 【版本】

python2 与 python3 是目前主要的两个版本。如下两种情况下,建议使用 python2:

- 1 你无法完全控制你即将部署的环境时;
- 2 你需要使用一些特定的第三方包或扩展时;

python3 是官方推荐的且是未来全力支持的版本,目前很多功能提升仅在 python3 版本上进行。

## 【hello world】

- 1 创建 hello.py
- 2 编写程序:

## Python 语言十分钟快速入门 II

```
if __name__ == '__main__':  
    print "hello word"
```

3 运行程序:

```
python ./hello.py
```

### 【注释】

1 无论是行注释还是段注释,均以 # 加个空格来注释。

2 如果需要在代码中使用中文注释,必须在 python 文件的最前面加上如下注释说明:

```
# -*- coding: UTF-8 -*-
```

3 如下注释用于指定解释器

```
#!/usr/bin/python
```

### 【文件类型】

1 Python 的文件类型分为 3 种,即源代码、字节代码和优化代码。这些都可以直接运行,不需要进行编译或连接。

2 源代码以 .py 为扩展名,由 python 来负责解释;

3 源文件经过编译后生成扩展名为 .pyc 的文件,即编译过的字节文件。这种文件不能使用文本编辑器修改。pyc 文件是和平台无关的,可以在大部分操作系统上运行。如下语句可以用来产生 pyc 文件:

```
import py_compile  
py_compile.compile('hello.py')
```

4 经过优化的源文件会以 .pyo 为后缀,即优化代码。它也不能直接用文本编辑器修改,如下命令可用来生成 pyo 文件:

```
python -O -m py_compile hello.py
```

### 【变量】

1 python 中的变量不需要声明,变量的赋值操作即使变量声明和定义的过程。

2 python 中一次新的赋值,将创建一个新的变量。即使变量的名称相同,变量的标识并不相同。用 id() 函数可以

获取变量标识:

```
x = 1  
  
print id(x)  
  
x = 2  
  
print id(x)
```

3 如果变量没有赋值,则 python 认为该变量不存在

4 在函数之外定义的变量都可以称为全局变量。全局变量可以被文件内部的任何函数和外部文件访问。

5 全局变量建议在文件的开头定义。

6 也可以把全局变量放到一个专门的文件中,然后通过 import 来引用:

gl.py 文件中内容如下:

```
_a = 1  
_b = 2
```

use\_global.py 中引用全局变量:

```
import gl  
  
def fun():  
  
    print gl._a  
    print gl._b  
  
fun()
```

### 【常量】

python 中没有提供定义常量的保留字。可以自己定义一个常量类来实现常量的功能。

```
class _const:  
  
    class ConstError(TypeError): pass  
    def __setattr__(self, name, value):  
        if self.__dict__.has_key(name):  
            raise self.ConstError, "Can't rebind const(%s)" % name  
        self.__dict__[name] = value  
  
    import sys  
    sys.modules[__name__] = _const()
```



## Python 语言十分钟快速入门 III

### 【数据类型】

1 python 的数字类型分为整型、长整型、浮点型、布尔型、复数类型。

2 python 没有字符类型

3 python 内部没有普通类型,任何类型都是对象。

4 如果需要查看变量的类型,可以使用 type 类,该类可以返回变量的类型或创建一个新的类型。

5 python 有 3 种表示字符串类型的方式,即单引号、双引号、三引号。单引号和双引号的作用是相同的。python 程序员更喜欢用单引号, C/Java 程序员则习惯使用双引号表示字符串。三引号中可以输入单引号、双引号或换行等字符。

### 【运算符和表达式】

1 python 不支持自增运算符和自减运算符。例如 i++/i-- 是错误的,但 i+=1 是可以的。

2 1/2 在 python2.5 之前会等于 0.5,在 python2.5 之后会等于 0。

3 不等于为 != 或 <>

4 等于用 == 表示

5 逻辑表达式中 and 表示逻辑与, or 表示逻辑或, not 表示逻辑非

### 【控制语句】

1 条件语句:

```
if (表达式):
```

```
    语句 1
```

```
else:
```

```
    语句 2
```

2 条件语句:

```
if (表达式):
```

```
    语句 1
```

```
elif (表达式):
```

```
    语句 2
```

```
...
```

```
elif (表达式):
```

```
    语句 n
```

```
else:
```

```
    语句 m
```

3 条件嵌套:

```
if (表达式 1):
```

```
    if (表达式 2):
```

```
        语句 1
```

```
    elif (表达式 3):
```

```
        语句 2
```

```
...
```

```
    else: 语句 3
```

```
elif (表达式 n):
```

```
...
```

```
else:
```

```
...
```

4 python 本身没有 switch 语句。

5 循环语句:

```
while(表达式):
```

```
...
```

```
else:
```

```
...
```

6 循环语句:

```
for 变量 in 集合:
```

```
...
```

```
else:
```

```
...
```

■ 本文未完,详细部分请参考原文

<http://developer.51cto.com/art/201207/347006.htm>

## 编者按

项目管理与软件开发的质量、效率、最终成果息息相关,本文主要讲述软件项目的风险评估、成本预算、客户沟通、需要分析、开发管理、成品交付等多个流程。

# 软件项目管理流程总结

项目管理与软件开发的质量、效率、最终成果息息相关,本文主要讲述软件项目的风险评估、成本预算、客户沟通、需要分析、开发管理、成品交付等多个流程。

在现今国内的项目的管理形式十分零乱,对管理欠缺重视,以致很多项目因为失去管理而最终折腰。

很多的实战形人才只重视于开发环节,而对其他的流程欠缺认识(包括本人),因而导致项目欠缺有条理的、阶段化的管理。

本人是一个典型的只重视开发的管理者,在多次的教训中深刻地体会到管理的重要性,因而以此文章对项目管理作出一个总结,当中存在很多的不足之处,敬请各位点评!

项目管理流程下载 :<http://developer.51cto.com/art/201207/346165.htm>

## 一、 风险评估

软件项目风险是指在整个项目周期中所涉及的成本预算、开发进度、技术难度、经济可行性、安全管理等各方面的问題,以及由这些问題而对项目所产生的影响。项目的风险与其可行性成反比,其可行性越高,风险越低。软件项目的可行性分为经济可行性、业务可行性、技术可行性、法律可行性等四个方面。而软件项目风险则分为产品规模风险、需要风险、相关性风险、管理风险、安全风险等六个方面:

### 1. 产品规模风险

项目的风险是与产品的规模成正比的,一般产品规模越大,问題就越突出。尤其是估算产品规模的方法,复用软件的多少,需求变更的多少等因素与产品风险息息相关:

- (1) 估算产品规模的方法

- (2) 产品规模估算的信任度
- (3) 产品规模与以前产品规模平均值的偏差
- (4) 产品的用户数
- (5) 复用软件的多少
- (6) 产品需求变更的多少

## 2. 需求风险

很多项目在确定需求时都面临着一些不确定性。当在项目早期容忍了这些不确定性,并且在项目进展过程当中得不到解决,这些问題就会对项目的成功造成很大威胁。如果不控制与需求相关的风险因素,那么就很有可能产生错误的产品或者拙劣地建造预期的产品。每一种情况对产品来讲都可能致命的,这些的风险因素有:

- (1) 对产品缺少清晰的认识
- (2) 对产品需求缺少认同
- (3) 在做需求分析过程中客户参与不够
- (4) 没有优先需求
- (5) 由于不确定的需要导致新的市场
- (6) 不断变化需求
- (7) 缺少有效的需求变化管理过程
- (8) 对需求的变化缺少相关分析等

## 3. 相关性风险

许多风险都是因为项目的外部环境或因素的相关性产生的。控制外部的相关性风险,能缓解策略应该包括可能性计划,以便从第二资源或协同工作资源中取得必要的组成部分,并觉察潜在的问题,与外部环境相关的因素有:

- (1) 客户供应条目或信息

## 软件项目管理流程总结 II

- (2) 交互成员或交互团体依赖性
- (3) 内部或外部转包商的关系
- (4) 经验丰富人员的可得性
- (5) 项目的复用性

### 4. 技术风险

软件技术的飞速发展和经验丰富员工的缺乏,意味着项目团队可能会因为技巧的原因影响项目的成功。在早期,识别风险从而采取合适的预防措施是解决风险领域问题的关键,比如:培训、聘请顾问以及为项目团队招聘合适的人才等。关于技术主要有下面这些风险因素:

- (1) 缺乏培训
- (2) 对方法、工具和技术理解的不够
- (3) 应用领域的经验不足
- (4) 对新的技术和开发方法应用不熟悉

### 5. 管理风险

尽管管理问题制约了很多项目的成功,但是不要因为风险管理计划中没有包括所有管理活动而感到惊奇。在大部分项目里,项目经理经常是写项目风险管理计划的人,他们有先天性的不足——不能检查到自己的错误。因而,使项目的成功变得更加困难。如果不正视这些棘手的问题,它们就很有可能在项目进行的某个阶段影响项目本身。当我们定义了项目追踪过程并且明晰项目角色和责任,就能处理这些风险因素:

- (1) 计划和任务定义不够充分
- (2) 对实际项目状态不了解
- (3) 项目所有者和决策者分不清
- (4) 不切实际的承诺
- (5) 不能与员工之间的进行充分地沟通

### 6. 安全风险

软件产品本身是属于创造性的产品,产品本身的核心技术保密非常重要。但一直以来,我们在软件这方面的安全意识比较淡薄,对软件产品的开发主要注重技术本身,而忽略了专利的保护。软件行业的技术人员流动是很普遍的现象,随着技术人员的流失、变更,很能会导致产品和新技术的泄密,致使我们的软件产品被它公司窃取,导致项目失败。而且在软件方面关于知识产权的认定目前还没有明确的一个行业规范,这也是我们软件项目潜在的风险。

### 7. 回避风险的方式

(1) 以开发方诱导能保证需求的完整,使需求与客户的真实期望高度一致。再以书面方便形成《用户需求》这一重要的文档,避免疏漏造成的损失在软件系统的后续阶段被逐步地放大。

(2) 设立监督制度,项目开发中任何较大的决定都必须有客户参与进行的,在该项目中项目监督由项目开发中的质量监督组来实施。

(3) 需求变更需要经过统一的负责人提出,并且要用户需求的审核领导认可,需求变更应该是定期而不是随时的提出,而且开发方应该做好详细的记录,让客户了解需求变更的实际情况。

(4) 控制系统的复杂程度,过于简单的系统结构,对用户来使用比例会有明显的折扣,甚至造成软件寿命过短。反之,软件结构的过于灵活和通用,必然引起软件实现的难度增加,系统的复杂度会上升,这又会在实现和测试阶段带来风险。适当控制系统的复杂程度有利于降低开发的风险。

## 软件项目管理流程总结 III

### 二、 成本预算

#### 1. 成本预算方式

##### (1) 自上而下的预算方法

自上而下的预方法主要是依据上层、中层项目管理人员的管理经验进行判断,对构成项目整体成本的子项目成本进行估计,并把这些判断估计的结果传递给低一层的管理人员,在此基础上由这一层的管理人员对组成项目的子任务和子项目的成本进行估计,然后继续向下一层传递他们的成本估计,直到传递到最低一层。

使用此预算方式,在上层的管理人员根据他们的经验进行的费用估计分解到下层时,可能会出现下层人员认为上层的估计不足以完成相应任务的情况。这时,下层人员不一定会表达出自己的真实观点,不一定会和上层管理人员进行理智地讨论,从而得出更为合理的预算分配方案。在实际中,他们往往只能沉默地等待上层管理者自行发现问题并予以纠正,这样往往会给项目带来诸多问题。

自上而下更适用于项目启动的前期,与真实费用相差在 30% ~ 70% 之间。

Scrum 使用自上而下的成本预算方式,它不会立即精确地确定成本,而是以最大限度容纳客户对未来产品要求所产生的变更。

##### (2) 自下而上的预算方法

自下而上方法要求运用 WBS (Work Breakdown Structure, 工作分解结构) 对项目的所有工作任务的时间和预算进行仔细考察。最初,预算是针对资源(团队成员的工作时间、硬件的配置)进行的,项目经理在此之上再加上适当的

间接费用(如培训费用、管理费用、不可预见费等)以及项目要达到的利润目标就形成了项目的总预算。自下而上的预算方法要求全面考虑所有涉及到的工作任务,更适用于项目的初期与中期,它能准备地评估项目的成本,与真实费用相差在 5% ~ 10% 之间。

注解: WBS

WBS 是面向提交成果对项目的分解,从提交成果的列表可以确定每个提交成果需要执行的活动。Scrum 会对 WBS 进一步细化,把一个迭代分解为一个或多个的工作包,再把工作包分解为细小的开发任务(一般开发任务的开发周期在 15 个工作小时以内)。

### 2. 确定项目支出

总体成本预算就是结合下列多个成本预算方式综合计算的开发成本:

#### (1) 零基数预算

在成本预算的初期应该使用零基数的计算原则,而不可以使用类似于:以上一年总体费用加上 20% 这样粗略的方式计算项目成本。

#### (2) 软硬件成本、物品成本

物品成本是指类似于:服务器(RAM 硬盘 CPU NIC 卡 RAID 簇)成本、维护成本、机房租金、光纤通讯成本、软件成本等的成本。

计算成本时需要考虑组装硬盘需时的长短,技术人员需要具备的素质,产品供应商能否提供质量保证,管理时是否需要额外的管理人员这些多方因素。

■ 本文未完,详细部分请参考原文

<http://developer.51cto.com/art/201207/346165.htm>



# 多线程程序与单进程程序的架构

多线程程序和单进程程序中都有使用,在网络服务中都有使用。多线程程序主要用于同步的处理一些请求,为了在同一时间处理较多的请求。一般采取线程池的策略,一个主线程来分配请求。

多线程程序和单进程程序中都有使用,在网络服务中都有使用。多线程程序主要用于同步的处理一些请求,为了在同一时间处理较多的请求。一般采取线程池的策略,一个主线程来分配请求。

## (一)常见的进程模型:

### (1)多线程模型

这种模式固定线程数量,吞吐量是接近每个线程的吞吐量 \* 线程数。不具有伸缩性,伸缩性不强,在高负载下能力有很大限制。对于负载不大的情况下的同步数据访问应用还是比较广泛的。

### (2)多线程模型二

这种模型是一个线程负责调度任务到不同的线程,通常维护一个线程池,线程池的线程数量可以伸缩,由调度线程决定。回报也通过调度线程来处理,线程与线程之间通过共享内存来获取其他方式来通信都可。但是其最高的线程数量也有限制。

### (1)单进程模型

这种模型采用全异步的模式处理请求,进程维护一个每个请求的全局队列,工作进程与网络进程是独立的,采用共享内存的模式进行通信,在全异步的情况下,也能做到高并发量,和高吞吐量。这种模型易于控制,需要对过载进行保护。

## (二)几种常见的模型对比

### (1)程序复杂度

最复杂是采用具有调度功能的线程池,这种模型需要一个调度线程,调度线程要做的工作包括指派工作线程,过载保护,与多个线程通信,同步。线程之间共享资源要加锁,线程之间可以消除共享资源,提高处理能力。可以采用同步的处理的模式访问,但是风险较大。

单进程的模型比较简单,一个进程将要完成所有功能,包括收包,逻辑处理,过载保护。

### (2)性能

一般来说,前端接收请求的,具有多客户端,多版本,多协议的适合采用多线程的模式,如果有统一的协议格式,则应当采用具有调度线程的线程池来完成。如果协议不同,则需要采用不同的线程监听不同的端口来解决。

后台服务一般适合采用单进程。后台线程要求高并发量,高吞吐量。单进程模型消除了锁,并且避免了多线程很多的复杂问题。多线程在开发过程中还有个很大的缺点是不利于调试。对开发者的调式技巧要求较高。

后台也有使用多线程的,后台的后……

■本文未完,详细部分请参考原文

<http://developer.51cto.com/art/201206/343688.htm>

## 如何写一份良好的缺陷（Bug）报告

没错,任何软件都存在 bug,哪怕是我们自己也存在缺陷,因为程序员也是普通人,人是会犯错误的。当有人在使用软件时遇到 bug,你需要使用邮件形成一份缺陷 bug,发送给开发人员。开发者可以依据该报告定位问题,复现问题,修复问题。

没错,任何软件都存在 bug,哪怕是我们自己也存在缺陷,因为程序员也是普通人,人是会犯错误的。当有人在使用软件时遇到 bug,你需要使用邮件形成一份缺陷 bug,发送给开发人员。开发者可以依据该报告定位问题,复现问题,修复问题。

但是很多时候,开发人员很难理解提交上的缺陷报告,因为发送人并不了解我们需要的是什麼,那如何与开发人员沟通以及如何写出一份清晰的缺陷报告,在这篇文章,我将教你如何写出一份清晰的缺陷报告能使开发者理解、复现、修复问题,这里下载缺陷报告模板。



### 为什么要发送缺陷报告

缺陷报告可以用很多方式来帮助我们的开发者。

- 他们能告知我们没有意识到的问题
- 他们能发现我们可能还没想到的新特性
- 他们能帮助我们感受到客户是如何使用

我们的软件,以至于我们可以做的更好

没有这些缺陷报告,我们就不知道出错的地方,我们需要它就像你唱歌跳舞时需要软件的支持一样。

什么时候发送缺陷报告

- 简单来说就是越快越好,详细来说就是:

● 当你看到一个错误消息时就发送错误报告

- 当屏幕是空白或者数据缺失就发送报告

- 当程序没有出现预期的结果时发送报告

● 当程序崩溃、死机、没有响应或者响应很慢时发送报告

- 当程序返回错误结果时发送报告

- 当你得不到想要的结果时发送报告

- 如果你不清楚怎样做时发送报告

● 如果你不喜欢软件做的方式,或者软件老打搅你时,发送错误报告

● 如果你想在系统中实现一个变通方案时发送报告

### 缺陷报告需要有哪些内容

缺陷报告应该包含很多信息,你提供的信息越多效果越好,对于开发者,就像我,提供一个纯文本文件模板给你填充然后邮件发给我,当然也有表格形式的,但是最期待你自己杜撰一份然后

## 如何写一份良好的缺陷( Bug ) 报告 II

发给我。下面是一些必须包括的部分以及如何写好每部分:

**标题:** 创建一个简短的标题, 让问题看起来更清晰。“应用崩溃”是一个很恼人的标题因为它没有足够的信息包括在这份报告里面。取而代之的是标题应该包含错误消息和消息码, 或者是结果的名称以及失败时你正在做的事情。例如: Error 402: 访问拒绝当点击“发送邮件”这个例子就提供了缺陷系统的上下文信息。

差: “程序崩溃”, “报错”, “Bug”

好: “从 ‘Kifu’ 中打印时 5C79 错误”, “Kifu honors’ 报表为空”

**产品:** 用名称标识产品, 告知你使用的是哪个版本。绝大部分软件都包含有版本信息。web 应用的版本信息通常在页脚。

差: “你的应用”

好: “Kifu v1.01”

**平台:** 告诉我们软件运行在什么平台。尤其是操作系统的名字及版本和浏览器名称版本。特别是 web 应用, 这些信息对我们很重要。

差: “Windows”

好: “Windows7, IE9”

**是否能重现:** 有些恼火的 Bug 是间歇性的出现, 我们想预先知道, 如果我们正在处理一个灵异事件或者正逢 Bug 出现时。

差: 留空白

好: “每次”, “偶然”, “不重现”

**描述:** 这部分是很多人拿不定的地方, 不知道怎么描述问题, 在描述中做到包括下面的内容:

● **总结:** 用简洁的语言概括出 Bug 出现时

你正在做的事情。从上下文开始, 在操作应用的哪个部分。聚焦在你做的时候软件做了什么?

差: “系统不能用了”

好: 在 “honor report” 页面单击 “打印按钮”, 但是报表是空的。

● **发生了什么:** 一步一步描述你做的事情当 bug 出现时, 为什么你认为是错误的。事无巨细, 打印出菜单的名称, 页面标题, 点击时的按钮或者链接的名称。做相同的操作是不是出现一样的错误。

差: “空白报表”

好: “点击 ‘File/Save as...’, ‘Save’ 对话框弹出, 然后点击 ‘OK’ 按钮, 但是文件没有保存”

● **错误时什么:** 如果错误消息出现时, 拷贝粘贴整个信息, 这样更有利于我们跟踪错误。

差: “有个错误, 点击它始终读不出”

好: “Error 403: 访问拒绝”

● **复现的步骤:** 如果你可以让 bug 重现, 那太好了, 这能提供很大的帮助。一步步描述如何重现次 bug。

差: “打印没法使用”

好: “从 ‘Honors Report’ 页面, 点击 ‘打印按钮’”

● **预期结果:** 描述你预期发生的结果当 bug 发生时, 这部分特别有用如果程序没有按照你期待的结果发生时, 因为它很诡异。

差: “我期待能正常工作”

好: “我期待能看到……”

■ 本文未完, 详细部分请参考原文

<http://developer.51cto.com/art/201207/347468.htm>

# 我对架构师的理解:如何成为一个合格的架构师

作者 /daoxuebao\_com

在讨论架构之前,我们先上道菜,青椒土豆肉丝,这道小菜味道还是不错的,自私点了,不考虑您是否喜欢,今天就上它了。

准备原材料:食用油、青椒、土豆、肉丝、大葱、香醋、鸡精和食盐。当然根据需要您可以再加入其他辅料。把青椒、土豆、肉片都切成丝,大葱切好,OK,一切准备就绪,开火,往锅里加油,等油热后,放切好的葱片,闻到葱香,放肉丝,稍微加些酱油,爆炒,接着放土豆丝和青椒丝,等八成熟,撒些鸡精和食用盐,出锅。

细心的读者可能发现,刚开始的时候我好像并没有准备酱油,是的,我确实没准备酱油,坦白的讲,有时候当锅里的油热的时候,我突然发现葱忘记洗了,更谈不上切成葱片了,此时我会匆匆忙忙的去洗,去切,甚至有时候慌里慌张的把手给切破。

通过我们做上面的一道菜,我们总结了以下几点:

总结 1: 巧妇难为无米之炊,我们要想做好这道菜,需要原材料;

总结 2: 这些原材料以时间为轴心他们彼此之间是有顺序关系的;

总结 3: 可能在某一步骤里,我们突然想添加些事先并没准备好的原材料;

总结 4: 一旦形成热油锅,似乎你要在这么短的时间内完成这些动作,做过饭的朋友更能体会到这句话。

言归正传,以软件的思想去考虑上面的业务

(事情),原材料,你可以理解为类库;顺序关系,你可以通过事件来描述;事先并没准备好的原材料,你可以通过接口(抽象类、虚函数等),让用户重载去实现;到这里你会发现,一旦打开“煤气”,去“引爆”预先设计的事件、接口,就好比多米诺骨牌一样一个接一个的传递下去,在某一时刻,它会检查是否放了“葱片”、是否放了“肉丝”,不好,“食用油”你就没放,还炒什么菜,抛出异常……

是的,上面就是框架,要想设计一个好的框架,看来我们首先要知道“青椒土豆丝”的做法,它大概需要哪些“原材料”,以及这些“顺序关系”该如何通过具体的语言去实现;当然了,要炒出“不同的菜”,具体的原材料和顺序关系又是不同的。下面通过分析几个大家比较熟悉的框架来更详细的说明。

MFC 框架:

MFC 中的框架思想采用了 MVC 的思想,其中 CWinApp 是全局型的,整个程序的引爆也是其在“搞鬼”,在其内部有指向文档模版的指针,而模板又囊括了视图、视图的管理者(就是那个 frame)和文档类,顺序关系是靠消息泵来推动。通过下面的调用关系可以看到各个类的“相互依存”(说明:下面的表摘自网络)

您可以试着联想“炒菜”的过程去思考上面的这张表,如果您真的理解了,我相信您会觉得他们之间没有什么区别(不过坦白来讲真的……

■本文未完,详细部分请参考原文

<http://developer.51cto.com/art/201206/342949.htm>



## ■ 编者按

其实我做前端的时间也不是很长,写这篇文章也是和一些初学的前端开发人员做一次经验的分享与交流,如果有错误的地方还需要大家谅解和指正。

## 关于前端开发的那些事

其实我做前端的时间也不是很长,写这篇文章也是和一些初学的前端开发人员做一次经验的分享与交流,如果有错误的地方还需要大家谅解和指正。

说起前端开发,我们就从一些让人打瞌睡的概念说起吧。

### 关于 WEB 标准

什么是 WEB 标准? 也许很多人会认为是 DIV+CSS。其实这是一个错误的说法,WEB 标准是一系列的技术规范,而目前主要包括 XHTML1.0、CSS2.1、DOM1.0 和 ECMAJavaScript。当然了,WEB 标准也不仅仅是一个规范,而是一系列规范的总称。而上面说的 DIV+CSS 是 XHTML1.1 结合 CSS2.0 的一种制作方法,严格来说应该是 XHTML+CSS。

而 WEB 标准是由三大块组成的。首先是结构,主要是对网页的数据对象进行整理分类,技术体现分别是 HTML, XML, XHTML。(在这里大家可能会问,HTM 不也是网页的一种吗? 是的,实际上 HTM 与 HTML 没有本质意义的区别,原因是为了兼容一些不能识别四位扩展名的老系统(win32),开发者为了满足 DOS 仅能识别 8+3 的文件名而已。);其次就是表现,用于没有结构化的信息显示以及视觉上的处理,分别包含为样式,

颜色,大小等等,主要应用于 CSS 上。最后一个就是行为,用于文档对象模型定义与交互的体现,分别是 DOM 与 MAscript 来完成。

### 关于布局

在这个问题之前,我先跟大家谈谈一个故人,那就是“table”,table 也就是表格。早期的网站大部分都是用 table 来布局的,准确来说应该是用表格来布局。因为如果用 table 来布局,这时我们就不能叫 table 了,因为它几乎能完成所有结构上的事情。那为什么不用 table 来布局呢? 这就是我们要谈的节约成本,提升速度了。

首先,table 的可维护性低,太多的 table 属性要修改起来让人眼花缭乱,比如 border="0",align="left" 等等,要修改得每个页面都要更改。第二,嵌套太过复杂,tr td 会让你找到吐。第三,不能重用。第四,添加交互效果太难。第五,加载速度过慢。等等,每个问题都无不牵扯到成本和速度的问题。

如果采用 XHTML+CSS 来布局,上面的几种问题完全可以解决。因为我们可以做到网页表现与内容分离,也就是用 XHTML 来作为数据的表现,而所有的样式表现将由 CSS 来全部控制。这样不仅结构清晰,页面体积也会大大的减小,浏览器兼容性更好。当然还有很多好处,比如有利

## 关于前端开发的那些事 II

于 SEO 优化,更容易被搜索引擎收录,样式重用,维护方便,错误修改页面调整更加快捷等等。所以 W3C 推荐使用 WEB 标准开发的原因相信大家也很清楚了。

### 关于优化规范

针对规范的问题在前段阿里技术嘉年华上跟淘宝前端开发专家嗷嗷交流过,嗷嗷认为,大家了解最多的是网站性能优化的 34 条黄金法则,我们称之为雅虎军规,但也不能说是规范,它只是一个建议。但不同的公司都有着不同的背景会导致每个公司都会加上自己的一些特殊的東西。所以让人感觉有很多,每个人出来讲都不一样了。其实最基础的我还是觉得雅虎军规,就是前人的方法总结了,可能由于时代在变,有些方案也会发生变化,所以要自己再去深入研究。

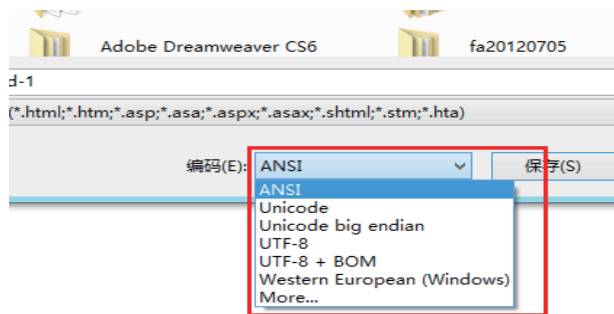
雅虎现在前端没有以前那么好,就是雅虎辉煌的时候很多人都去看看人家是怎么做的。现在雅虎就没有像以前那么重视了,或者是由于公司的一些原因。我们可能也不是太了解,但是他们曾经留下那些经验是非常好的。

### 关于字符集

在每个页面的 <head> 中我们都有设置了字符集,而在通常的情况下我们用最多的时"UTF-8(国际标准编码)"和"GB2312(中文编码)".一般来说,在前端表现上面我们用 UTF-8,后台 cms 我们就使用 GB2312。

<meta http-equiv="Content-Type" content="text/html; charset=GB2312" /> 有时候我们在打开一些网页上会显示口口口口 ??? 这样的字符,可能是你在在写 charset 的时候用了欧洲

语系,比如 ISO-8859-1 等等,导致浏览器无法辨别语种;再者是因为你的页面编码不对,这时我们要通过保存来改变页面编码。可以通过记事本或者 EditPlus 操作:



在这里主要介绍三个编码(部分内容借鉴汪旭的一篇帖子):

**ANSI 编码:** 简单的说就是把所有的空格、标点符号、数字、大小写字母分别用连续的字解状态表示,一直编到第 127 号,这样计算机就可以用不同字节来存储英语的文字了。所以人们就把这个方案叫做 ANSI 的“ASC II”编码。这个肯定是不适合在国内使用的。

**GB2312 编码:** 这个编码的出现也是计算机普及到我们中国时,这时我们的汉字如何通过计算机表现出来呢? 6000 多个常用的汉字如何去保存呢? 其实也就是通过两个大于 127 的字符连载一起是,就表示一个汉字。前面的一个字节(他称之为高字节)从 0xA1 用到 0xF7,后面一个字节(低字节)从 0xA1 到 0xFE,这样我们就可以组合出大约 7000 多个简体汉字了。

本文未完,请阅读原文: ■

<http://developer.51cto.com/art/201207/345418.htm>